465 6 APR 73

NAVAL RESEARCH LOGISTICS QUARTERLY

MARCH 1973
VOL. 20, NO. 1

OFFICE OF NAVAL RESEARCH

NAVSO P-1278

NAVAL RESEARCH LOGISTICS QUARTERLY

EDITORS

F. D. Rigby
Texas Tech. University

O. Morgenstern
New York University

B. J. McDonald
Office of Naval Research

S. M. Selig
Managing Editor
Office of Naval Research
Arlington, Va. 22217

ASSOCIATE EDITORS

R. Bellman, RAND Corporation
J. C. Busby, Jr., Captain, SC, USN (Retired)
W. W. Cooper, Carnegie Mellon University
J. G. Dean, Captain, SC, USN
G. Dyer, Vice Admiral, USN (Retired)
P. L. Folsom, Captain, USN (Retired)
M. A. Geisler, RAND Corporation
A. J. Hoffman, International Business Machines Corporation
H. P. Jones, Commander, SC, USN (Retired)
S. Karlin, Stanford University
H. W. Kuhn, Princeton University
J. Laderman, Office of Naval Research
R. J. Lundegard, Office of Naval Research
W. H. Marlow, The George Washington University
R. E. McShane, Vice Admiral, USN (Retired)
W. F. Millson, Captain, SC, USN
H. D. Moore, Captain, SC, USN (Retired)
M. I. Rosenberg, Captain, USN (Retired)
D. Rosenblatt, National Bureau of Standards
J. V. Rosapepe, Commander, SC, USN (Retired)
T. L. Saaty, University of Pennsylvania
E. K. Scofield, Captain, SC, USN (Retired)
M. W. Shelly, University of Kansas
J. R. Simpson, Office of Naval Research
J. S. Skoczylas, Colonel, USMC
S. R. Smith, Naval Research Laboratory
H. Solomon, The George Washington University
I. Stakgold, Northwestern University
E. D. Stanley, Jr., Rear Admiral, USN (Retired)
C. Stein, Jr., Captain, SC, USN (Retired)
R. M. Thrall, Rice University
T. C. Varley, Office of Naval Research
J. F. Tynan, Commander, SC, USN (Retired)
J. D. Wilkes, Department of Defense OASD (ISA)

The Naval Research Logistics Quarterly is devoted to the dissemination of scientific information in logistics and will publish research and expository papers, including those in certain areas of mathematics, statistics, and economics relevant to the over-all effort to improve the efficiency and effectiveness of logistics operations.

Information for Contributors is indicated on inside back cover.

The Naval Research Logistics Quarterly is published by the Office of Naval Research in the months of March, June, September, and December and can be purchased from the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. Subscription Price: $10.00 a year in the U.S. and Canada, $12.50 elsewhere. Cost of individual issues may be obtained from the Superintendent of Documents.

The views and opinions expressed in this quarterly are those of the authors and not necessarily those of the Office of Naval Research.

Issuance of this periodical approved in accordance with Department of the Navy Publications and Printing Regulations NAVEXOS P-35

Permission has been granted to use the copyrighted material appearing in this publication.

STRUCTURED MARKOVIAN DECISION PROBLEMS

Basil A. Kalymon*

Faculty of Management Studies
University of Toronto

ABSTRACT

Structured finite action-finite state space discounted Markovian decision problems are analyzed. Any problem of a general class is shown to be equivalent to a "separated" problem with decomposable problem structure. A modified policy iteration approach is developed for this decomposable reformulation. Both analytic and computer evaluations of the decomposition algorithm's effectiveness are presented.

INTRODUCTION

In this paper we study a class of discounted, infinite horizon, finite-action, finite-state space Markovian decision problems which, when appropriately reformulated, have decomposable structure. Reformulation is accomplished by using an extension of the concept of separability discussed in [2], [4], and [11]. We give the general form of such problems and show their equivalence to the derived decomposable problem. Applications of such separation to multi-period models involving two dimensional stochastic processes and other areas of potential application are discussed.

To exploit the structure of the derived problem, we develop a policy iteration algorithm based on the concept of efficient decomposition of the required matrices, and indicate how value iteration might be used intermittently (as suggested in [3]). Applicability of the algorithm to other more general structured problems is discussed. Further, we analyze computational efficiency and present some numerical results.

Previous works in this area (see [4] for summary) was concerned primarily with the separability of the problem and did not consider the possibility of a resulting highly structured problem. The development below shows how zero-time transitions to some additional special states can separate a problem so as to reveal its decomposable structure. Hence, the efficient, specialized algorithm can be applied. This specialization is related to similar work on structured linear programs† as studied in [6]. The algorithm, in essence, exploits the special nature of Markovian decision problems to modify and simplify an applicable, structured policy iteration algorithm.

Having presented the general form of the problems of interest in sec. 1, we derive the equivalent separated problem, and discuss applications. Section 2 deals with decomposition algorithms which

* This research was partially supported by the National Science Foundation under Grant GP–26294 and was conducted while the author was at the University of California, Los Angeles.

† The equivalence of policy iteration to application of multiple substitution in the dual simplex method of a related linear program is known (see [5] or [2]).

may be applied to the reformulated problem. Analytical and numerical evaluations of the proposed computational method are presented in the final sec. 3.

1. SEPARATION OF THE GENERAL STRUCTURED MODEL

The general Markovian decision problem to be analyzed shall consist of a finite number of states labeled (i, j) with $1 \leq i \leq N(j)$, $1 \leq j \leq M$. Using the notation $(i, \cdot) \equiv \{(i, j) : 1 \leq j \leq M\}$ and $(\cdot, j) \equiv \{(i, j) : 1 \leqq i \leq N(j)\}$, we assume that for any $(i, j) \epsilon (i, \cdot)$ we may take any action $k \epsilon A_{i,j}$. Given that an action k is taken at (i, j), an expected cost of $c_{i,j}^k$ is incurred and transition to state (m, l) in the next period occurs with probability $F_{(i,j),(m,l)}^k$. Assuming that future costs are discounted at the rate $0 < \alpha < 1$, we seek a policy which minimizes the expected total discounted costs over an infinite planning horizon. The above problem shall be denoted as Problem A.

This problem, so far, is identical to the general finite-action, finite state-space Markovian decision problem studied by Howard in [7]. The designation of the states by a two-dimensional index facilitates the discussion of the problem structure to be introduced below. The sets of states $(i, \cdot)$ and $(\cdot, j)$ provide convenient partitions of the entire state space into sections which share similar exploitable characteristics.

The special assumptions on the structure of Problem A which make it separable are that $A_{i,j} = \{ \bigcup_{i' \epsilon S(i,j)} L_{i'} \} \cup K_{i,j}$ where $S(i, j)$ is some subset of $\{1, 2, \ldots, N(j)\}$. Given any action $k \epsilon L_{i'}$ the cost and transition probabilities can be expressed as

$$c_{i,j}^k = a_{i,j}^{i'} + b_{i'}^k \tag{1-1}$$

$$F_{(i,j),\cdot}^k = F_{i',\cdot}^k,$$

where the $\cdot$ notation refers to the entire row vector of probabilities. The assumptions in (1–1) imply that, given that an action in $L_{i'}$ is to be chosen and the price $a_{i,j}^{i'}$ has been paid, the subsequent costs and transition probabilities are independent of the particular state (i, j) at which the action occurs, but depend only on i' and k. Restrictions on the actions in $K_{i,j}$ which result in further simplifications shall be indicated below in (1–2). The conditions under which the proposed decomposition is efficient shall be discussed in detail in sec. 3, but, intuitively, the number of actions in $K_{i,j}$ which "link" (i, j) with states outside of $(\cdot, j)$ should be small relative to the size of $A_{i,j}$. Also, the number M of subsets $(\cdot, j)$ should be at least as large as the size, say, N of the largest set of states $(i, \cdot)$ (N is defined by $N = \max \{N(j) \mid 1 \leq j \leq M\}$).

Under the conditions of (1–1), we now proceed to define a second problem, designated as Problem B. Problem B has the same states as Problem A plus an additional set of states $(\cdot, M+1) \equiv \{(i, M+1) : 1 \leq i \leq N\}$. At any of the original states (i, j), we may take any action $k \epsilon \{t_{i'} : i' \epsilon S(i, j)\} \cup K_{i,j}$ where $t_{i'}$ is an action which at a cost of $a_{i,j}^{i'}$ causes an instantaneous (zero-time) transition to the added state $(i', M+1)$. At any added state $(i, M+1)$, we may take any action $k \epsilon L_i$ at a cost b_i^k and with the resulting transition probability vector $F_{i,\cdot}^k$ over states in the next period. In the above, it is assumed that $F_{(i,j),(l,m)}^k$ ($F_{i,(l,m)}^k$) is equal to zero where previously undefined (i.e., at $(l, m) \epsilon (\cdot, M+1)$).

Comparing Problem B with Problem A we see that N states have been added, giving a total of $\sum_{j=1}^{M} N(j) + N$ states in Problem B. However, the number of actions at an original state (i, j) has been

reduced from $\sum_{i' \in S(i,j)} |L_{i'}| + |K_{i,j}|$ to $\sum_{i' \in S(i,j)} 1 + |K_{i,j}|$ (where $|G|$ denotes the cardinality of the set G). Thus, the total number of state-action pairs at the original states (i.e., for $j \leq M$) has been reduced. Assuming that the $N(j)$ are approximately equal, and that $|L_i| >> 1$ would imply that the reduction is a significantly large number relative to the total state-action pairs in Problem A. The number of additional state-action pairs due to any added states $(i, M+1)$ would be $\sum_{i'} |L_{i'}|$. Thus, assuming that N is small relative to $\sum_{j=1}^{M} N(j)$, the number of state-action pairs (which determine the amount of computational effort required per iteration) will be significantly smaller in Problem B than in Problem A. Hence, we see first possible source of simplification from redefinition.

Further, we assume that $K_{i,j} = K^S_{i,j} \cup K^L_{i,j}$ with $|K^L_{i,i}| << |K^S_{i,j}|$, and for $k \in K^S_{i,j}$ we have

$$F^k_{(i,j),(l,m)} = 0 \qquad \text{for } m \neq j. \tag{1-2}$$

These conditions imply that the reformulated Problem B will be highly structured in the sense that only at state-action pairs $(i, j) - k$ with $k \in K^L_{i,j}$ or at $(i, j) \in (\cdot, M+1)$ would a one-step transfer to all other states in the problem be possible. At other state-action pairs one-step transfers only to states either in $(\cdot, j)$ or in $(\cdot, M+1)$ would be feasible. This structure implies that decomposition techniques can be applied, as shall be shown in sec. 3.

To establish the equivalence between Problems A and B, the notation $V_{i,j}$ and $V'_{i,j}$ shall represent the minimum expected total discounted costs over the infinite horizon (given the initial state (i, j)) in Problems A and B, respectively. Also, $V^{\delta}_{i,j}$ and $V^{\delta'}_{i,j}$ shall represent the infinite horizon costs of using stationary policies δ and δ' in Problems A and B, respectively (where $\delta_{i,j}$ ($\delta'_{i,j}$) is the action to be taken at (i, j)). These functions all exist and are the unique fixed points of appropriate 1-stage and 2-stage contraction mappings (see [3]). It is known (see [3]) that optimal* stationary policies δ and δ' exist.

Define policy δ as being "essentially equivalent" to policy δ' if for all (i, j) with $j \neq M+1$ we have

$$\delta_{i,j} = \begin{cases} \delta'_{i,j} & \text{for } \delta'_{i,j} \in K_{i,j} \\ \delta'_{i',M+1} & \text{for } \delta'_{i,j} = i'. \end{cases} \tag{1-3}$$

The following theorem establishes the equivalence of the two problems.

THEOREM 1.1: (i) For all states (i, j) which are in both Problem A and B (i.e., (i, j) with $j \neq M+1$), we have

$$V_{i,j} = V'_{i,j}.$$

(ii) For any optimal policy δ' for Problem B, the essentially equivalent policy δ is an optimal policy for Problem A.

Before proving Theorem 1.1, we establish the following:

LEMMA 1.1: If two policies δ and δ' are essentially equivalent, then for all (i, j) with $j \neq M+1$, we have

* Optimality is defined *w.r.t.* the set of all history remembering randomized policies.

(1–4) $$V^{\delta}_{ij} = V^{\delta'}_{i,j}.$$

PROOF: Considering v and v' as column variable vectors of appropriate dimensions, $V^{\delta'}$ is the unique solution to the following equations:

(1–5) $$v'_{i,j} = c^{k}_{i,j} + \alpha F^{k}_{(i,j),\cdot}\, v' \qquad \text{if } k \equiv \delta'_{i,j} \epsilon K_{i,j} \text{ for } 1 \leq i \leq N(j),\ 1 \leq j \leq M;$$

(1–6) $$v'_{i,j} = a^{i'}_{i,j} + v_{i',M+1} \qquad \text{if } \delta'_{i,j} = t_{i'} \text{ for } 1 \leq i \leq N(j),\ 1 \leq j \leq M;$$

(1–7) $$v'_{i,M+1} = b^{k}_{i} + \alpha F^{k}_{i,\cdot} v' \qquad k \equiv \delta'_{i,M+1} \text{ for } 1 \leq i \leq N.$$

But, by substitution of Equations (1–7) in (1–6), it is immediate by (1–1) and the definition of equivalence (1–3) that the $\{V^{\delta'}_{i,j} : j \neq M+1\}$ satisfy the equations

(1–8) $$v_{i,j} = c^{k}_{i,j} + \alpha F^{k}_{(i,j),\cdot} v \qquad k \equiv \delta_{i,j} \text{ for } 1 \leq i \leq N(j) \text{ and } 1 \leq j \leq M.$$

But, Equations (1–8) are uniquely satisfied by V^{δ} and hence (1–4) holds as required.

PROOF OF THEOREM 1.1: First, we shall establish Part (ii). Given that $\delta'_{i,j}$ is an optimal action in Problem B, we know that for $j \neq M+1$

(1–9) $$V^{\delta'}_{i,j} = \min\left\{ \min_{i' \epsilon S(i,j)} \left\{ a^{i'}_{i,j} + V^{\delta'}_{i',M+1} \right\}, \min_{k \epsilon K_{i,j}} \left\{ c^{k}_{i,j} + \alpha F^{k}_{(i,j),\cdot}\ V^{\delta'} \right\} \right\},$$

where

(1–10) $$V^{\delta'}_{i,M+1} = \min_{k \epsilon L_i} \left\{ b^{k}_{i} + F^{k}_{i,\cdot}\, V^{\delta'} \right\}.$$

Thus, from (1–1), (1–9), and (1–10), we get

(1–11) $$V^{\delta'}_{i,j} = \min_{k \epsilon A_{i,j}} \left\{ c^{k}_{i,j} + \alpha F^{k}_{(i,j),\cdot} \bar{V}^{\delta'} \right\},$$

where $\bar{V}^{\delta'}$ is the section of $V^{\delta'}$ for which $j \neq M+1$. But, by the essential equivalence of δ and δ', applying Lemma 1.1, from (1–11) we get

(1–12) $$V^{\delta}_{i,j} = \min_{k \epsilon A_{i,j}} \left\{ c^{k}_{i,j} + \alpha F^{k}_{(i,j),\cdot}\, V^{\delta} \right\}.$$

This implies that δ is an optimal policy for Problem A, establishing Part (ii) of the theorem.

Part (i) follows immediately from Part (ii) and Lemma 1.1, thus completing the proof of Theorem 1.1. Q.E.D.

Theorem 1.1 enables us to analyze a simplified Problem B instead of the original Problem A. As an application of this type of reduction, consider any "mixed-Markovian" problem containing two separate controlled stochastic processes in which one of the processes is independent from period to period. In this case, the state space can be naturally described by a two dimensional representation (i, j) where the i represents the state of the first process and the j the state of the independent process. If the problem is such that the j process affects the costs of actions at a given i only in an additive manner, then we arrive at a problem in which essentially the same problem (with only cost differences) must be solved for each different level of j. The similarity of the duplicated problems can then be effectively exploited by applying the above reformulation procedures which simplify the problem and (as shall be demonstrated below) make the problem amenable to decomposition procedures. These (and similar) types of problems were found in introducing stochastic cost parameters in inventory, bond-refunding and machine replacement models and the possibilities of separation and decomposition were noted (e.g., see [8], [9] and [10] for specific and detailed formulations).

For example, consider a machine replacement model with stochastic Markovian deterioration and with cost of replacement in any period represented as an independent random variable. Assuming a finite number of deterioration states N and a finite number of cost states M, the states of the system are defined by $\{(i, j): 1 \leq i \leq N,\ 1 \leq j \leq \mathrm{M}\}$. If keeping the current machine ($k=1$) or replacing ($k=2$) are the only two possible actions at each state, then let

$$A_{i,j} = \{1,2\} = L_i \cup L_0,\ K_{i,j} = \phi$$

where $L_i = \{1\}$, $L_0 = \{2\}$ for $1 \leq i \leq N$. Now, considering $a^1_{i,j}$ ($a^2_{i,j}$) as the current period costs of keeping (replacing) the current machine at state (i, j), we can set $b^k_{i'} = 0$. By the assumptions of independence of costs

$$F^k_{(i,j),\cdot} = F^k_{i',\cdot}$$

where $i' = i$ or $i' = 0$ and $S(i, j) = \{0, i\}$.

In the above example, the number of state-action pairs in the reformulated problem would be $(2N \cdot M + N + 1)$ as compared to $(2N \cdot M)$ in the original problem, a slight increase. However, instead of dealing with a dense problem, the reformulated problem has only $N + 1$ state-action pairs (out of $2N \cdot M + N + 1$) in which transfer to all other states is possible in a single step and hence the decomsition procedures to be given in sec. 3 will be applicable.

This completes our discussion of separation and we shall now show what decomposition procedures may be applied to the reformulated problem.

2. DECOMPOSITION OF THE REFORMULATED PROBLEM

In discussing decomposition procedures, we shall refer to the linear programming problem which can be solved to find the optimal cost and policy (see [5], [2] or [3]). Figure 2.1 below gives the detached coefficient tableau for the linear optimality conditions for the separated problem (B), with the notation

defined as follows:

(2–1)

$v_j =$ $(v_{1,j}v_{2,j}, \ldots, v_{N(j),j}) \equiv$ the vector of variables corresponding to the states in $(\cdot, j)$.

$A_j =$ matrix of constraints due to actions $k \epsilon K^S_{i,j}$ at $(i, j) \epsilon (\cdot, j)$; there is one row of A_j corresponding to each state-action pair $(i, j) - k$ and it is given by $e_i - \alpha F^k_{(i,j),\cdot}$ where e_i is the unit row vector with the 1 in the ith column and $F^k_{(i,j),\cdot}$ is understood to be a row vector restricted to the subgroup $(\cdot, j)$.

$G_j, J_j =$ matrices defining constraints due to actions $t_{i'}$ (with $i' \epsilon S(i, j)$) at $(i, j) \epsilon (\cdot, j)$; there is one row of G_j and of J_j corresponding to each state-action pair $(i, j) - t_{i'}$ (where $i' \epsilon S(i, j)$) and the rows are given simply by e_i and $- e_{i'}$, respectively.

$(C_1, C_2, \ldots, C_M) \equiv C =$ matrix of constraints due to those actions $k \epsilon K^L_{i,j}$ at any (i, j); there is one row in C for every such $(i, j) - k$ pair and it is given by $e_{i,j} - \alpha F^k_{(i,j),\cdot}$ where $e_{i,j}$ is the unit row vector of appropriate dimension with the 1 in the column corresponding to $v_{i,j}$.

$(D_1, D_2, \ldots, D_M, E) \equiv (DE) =$ matrix of constraints due to those actions $k \epsilon L_{i'}$ for any $1 \leq i' \leq N$ for $(i, j) \epsilon (\cdot, M + 1)$; there is one row in D and E for each state-action pair $(i', M + 1) - k$ with $k \epsilon L_{i'}$ and they are given by $- \alpha F^k_{i',}$ and $e_{i'}$, respectively.

$\bar{1} \equiv$ constant row vector of 1's.

$c_j =$ vector of costs $c^k_{i,j}$ corresponding to taking action $k \epsilon K^S_{i,j}$ at state $(i, j) \epsilon (\cdot, j)$.

$a_j =$ vector of partial costs $a^{i'}_{i,j}$ corresponding to taking the action $t_{i'}$ at state $(i, j) \epsilon (\cdot, j)$.

$b =$ vector of partial costs $b^k_{i'}$ corresponding to taking the particular action $k \epsilon L_{i'}$ at state $(i', M + 1)$.

$z =$ objective variable to be maximized.

$$
\begin{array}{cccccccc}
v_1 & v_2 & \cdots & v_M & v_{M+1} & & & \\
\bar{1} & \bar{1} & & \bar{1} & \bar{1} & = & z & \\
A_1 & & & & & \leq & c_1 & \\
G_1 & & & & J_1 & \leq & a_1 & \\
 & A_2 & & & & \leq & c_2 & \\
 & G_2 & & & J_2 & \leq & a_2 & \\
 & & \ddots & & & & \vdots & \\
 & & & A_M & & \leq & c_M & \\
 & & & G_M & J_M & \leq & a_M & \\
C_1 & C_2 & \cdots & C_M & & \leq & c_{M+1} & \\
D_1 & D_2 & \cdots & D_M & E & \leq & b &
\end{array}
$$

Figure 2.1

We notice from Figure 2.1, that the separated problem exhibits block diagonal structure with linking variables and constraints. This structure appears only after separation, as the tableau for Problem A would be essentially dense. Of course for efficient application of decomposition procedures, the number of rows in C and D must be small relative to the rest of the problem. This will be true under the conditions on the problem structure discussed in sec. 1. The dual of the L.P. problem in Figure 2.1 is precisely of the form analyzed by Hartman and Lasdon in [6] and the generalized upper bounding algorithm developed in [6] would be applicable in solving Problem B.

As in generalized upper bounding L.P. algorithms, the structure of the problem may also be exploited in policy iteration. Specifically, the evaluation of any given policy requires the inversion of the policy matrix which corresponds to the basic matrix in L.P. For any policy δ for Problem B, define the associated "policy matrix" as $B_\delta = (I - P_\delta^\alpha)$, where P_δ^α consists of the appropriately discounted transition probability matrix associated with policy δ. For policy iteration in the Markovian decision problems considered, the following lemmas will be required.

LEMMA 2.1: Let $B_\delta = (I - P_\delta^\alpha)$ be the policy matrix associated with any policy δ for Problem B. Then, B_δ is nonsingular.

PROOF: P_δ^α for any policy δ for Problem B can be shown to be of the form

$$P_\delta^\alpha = \begin{pmatrix} \alpha P_\delta^1 & \alpha P_\delta^2 & 0 \\ 0 & 0 & P_\delta^3 \\ \alpha P_\delta^4 & \alpha P_\delta^5 & 0 \end{pmatrix}, \tag{2-2}$$

where $(\alpha P_\delta^1 \alpha P_\delta^2 0)$ refers to states (i, j) with $j \neq M+1$ for which actions in $K_{i,j}^S$ (i.e., not requiring a transfer to a "special" state) are chosen by δ, $(0\ 0\ P_\delta^3)$ corresponds to states (i, j) with $j \neq M+1$ for which actions $t_{i'}$, for $i' \epsilon S(i, j)$ (i.e., requiring immediate transfer to a "special" state) are chosen by δ and $(\alpha P_\delta^4 \alpha P_\delta^5 0)$ refers to the actions at the "special" states $(\cdot, M+1)$. That $(P_\delta^\alpha)^n \to 0$ as $n \to \infty$ is readily established since $0 < \alpha < 1$, and since $|P_\delta^i| \leq 1$ for $1 \leq i \leq 5$, and for any δ since it is a submatrix of a probability transition matrix. Thus, $(I - P_\delta^\alpha) = B_\delta$ is invertible (see Thm. 1.11.1 in [12]). Q.E.D.

LEMMA 2.2: By permutation of columns associated with (i, j) for $j \leq M$, a policy matrix B_δ associated with a policy δ for Problem B can be partitioned in the following manner

$$B_\delta = \begin{pmatrix} B_1 & & & & & H_1 \\ & B_2 & & & & H_2 \\ & & \cdot & & & \cdot \\ & & & \cdot & & \cdot \\ & & & & \cdot & \cdot \\ & & & & B_M & H_M \\ V_1 & V_2 & \cdot & \cdot & V_M & E^0 \end{pmatrix},$$

where each B_j is a square matrix. Then (if dimension of $B_j \neq 0$), B_j is nonsingular for each $j = 1, 2, \ldots, M$.

PROOF: First note that from Fig. 2.1 it is evident that B_δ can be partitioned in this manner,

simply by considering columns associated with states (i, j) for which actions $\delta_{i,j}$ are defined by the $(C_1, C_2, \ldots, C_M)$ rows as being part of the $(H_1, H_2, \ldots, H_M, E^0)$ section of B.

Next, since $B_\delta = (I - P_\delta^\alpha)$, it is immediate that $B_j = (I_j - P_{\delta_j}^\alpha)$, where I_j is an identity matrix of appropriate dimension and $P_{\delta_j}^\alpha$ is of the form

$$P_{\delta_j}^\alpha = \begin{bmatrix} \alpha P^1{}_{\delta_j} \\ 0 \end{bmatrix}, \tag{2-3}$$

where the rows of $P_{\delta_j}^1$ correspond to some (appropriately abbreviated) rows of A_j and the 0 to some rows of G_j. But, since $0 < \alpha < 1$ and since the rows of $P_{\delta_j}^1$ are merely segments of probability vectors, it is immediate that $(P_{\delta_j}^\alpha)^n \to 0$ as $n \to \infty$. Hence (providing that the dimension of $B_j \neq 0$), $B_j = (I_j - P_{\delta_j}^\alpha)$ is invertible (see Thm, 1.11.1 in [12]). Q.E.D.

LEMMA 2.3: The matrix Q defined by

$$Q = \left(E^0 - \sum_{j=1}^{M} V_j B_j^{-1} H_j\right)$$

is invertible.

PROOF: B_δ (with columns appropriately permuted) is nonsingular by Lemma 2.1. By Lemma 2.2, B_j^{-1} exists. Hence multiplying each block of rows of B_δ associated with B_j by $V_j B_j^{-1}$ and subtracting the result from the block of rows $(V_1, V_2, \ldots, V_M, E^0)$ we obtain the matrix

$$\begin{bmatrix} \bar{B}_\delta & H \\ 0 & E^0 - V\bar{B}_\delta^{-1} H \end{bmatrix}, \tag{2-4}$$

where V, H, and $\bar{B}_\delta$ have the obvious deflnitional correspondence to $\{V_j\}$, $\{H_j\}$, and $\{B_j\}$. Since the matrix of (2-4) is still nonsingular, $E^0 - V\bar{B}_\delta^{-1} H = E^0 - \sum_{j=1}^{M} V_j B_j^{-1} H_j = Q$ must be nonsingular since $(0 E^0 - V\bar{B}_\delta^{-1} H)$ must be linearly independent rows. Q.E.D.

The above lemma shows that the policy evaluation step of policy iteration may be performed in the following manner:

(i) Permute B_δ to achieve the structure of Lemma 2.2.

(ii) Invert the B_j's and hence, evaluate $Q = (E^0 - V\bar{B}_\delta^{-1} H)$.

(iii) Determine the values of V^δ associated with the columns of E° by solving $QV_{(E^\circ)}^\delta = C_{\delta(E^\circ)}$, where $V_{(E^\circ)}^\delta$ and $C_{\delta(E^\circ)}$ represent the appropriate subset of variables and costs, respectively.

(iv) For each j, determine the values of $V_{\cdot,j}^\delta$ associated with the columns of B_j by multiplying the appropriate vector of costs by B_j^{-1}.

Note that in the proof of Lemma 2.2, we give a simple means of permuting the columns of B_δ to achieve the diagonal structure. This points out a simplification possible in the problem at hand which is not achievable in the general linear programming problem of similar structure as discussed in [6].

The efficiency of the above policy iteration algorithm, except for the simplification noted above, should be comparable to that of the Hartman-Lasdon [6] decomposition procedure for L.P.'s. Further, improvements over the L.P. approach might be achieved by applying a modified policy iteration approach as suggested in [3]. Specifically, to avoid the computational effort required for policy evaluation, we might perform several iterations of policy improvement, through value iteration of the functional equation, for each policy evaluation performed. In the final section, the relative effectiveness of the above modified decomposed policy iteration approach as compared to a direct policy iteration approach to Problem A is analyzed. Also, computer running times for some sample problems are given.

3. ANALYSIS OF RELATIVE EFFICIENCY

We shall first discuss analytical evaluations of efficiency and shall then give computational results for some test problems. For simplicity, in this section we shall assume that each block $j(=1, 2, \ldots, M+1)$ has the same number of states N and also that at each state there is an equal number of actions K. Also, it is assumed that the set of (state-action) constraints defined by $(C_1, C_2, \ldots, C_M)$ is void.* Under the above conditions, by a count of the elementary additions and multiplications required,† we arrived at the expressions in Table 2.1 of the total number of elementary operations required per iteration for the indicated step. The estimate for the policy evaluation step includes the effort required to solve for V^δ from $B_\delta V^\delta = C_\delta$ (where C_δ is the vector of costs associated with policy δ) and, in the decomposed approach, includes the effort required to evaluate the matrix Q as well as the inversions of the B_j's and the subsequent evaluation of V^δ. The estimate of R performances of value iteration includes the effort required to determine the best replacement action at each state and the new set of values. Since the number of iterations taken will be identical whether the problem is decomposed or not, the expressions given in Table 2.1 reflect relative efficiencies of a direct and decomposition approach. We see from Table 2.1 that for the decomposition approach the effort will grow linearly in M as opposed to the direct approach for which effort will grow as a factor of M^3.

TABLE 2.1

	Policy iteration applied to Problem A	Decomposed policy iteration applied to Problem B
Policy evaluation step	$\simeq M^3N^3 + 3M^2N^2$	$\simeq (6M+2)N^3 + (5M+4)N^2$
Policy improvement step (R times)	$\simeq R(4M^2N^2K)$	$\simeq R(4MN^2K)$
Total per iteration	$\simeq M^3N^3 + (4RK+3)M^2N^2$	$\simeq (6M+2)N^3 + (4RK+5)MN^2 + 4N^2$

Table 2.2 presents computer running times for several test problems solved by the modified decomposition approach, using the indicated alternative strategies. The problems were generated both in a random manner with costs increasing monotonically (from subgroup to subgroup), and in an essentially random manner with a single favourable state in each subgroup. The starting policy

*If $(C_1, \ldots, C_M)$ is a large set of constraints, then it is obvious that a policy matrix B_δ would not possess significant structure and hence our decomposition procedure would not be efficient. Thus, assuming that it is void approximates the conditions under which our algorithm might be applied.

†We assume that inversion of a $(n \times n)$ matrix requires $\simeq (2n^3+2n^2)$ operations while addition (multiplication) of two $(n \times n)$ matrices requires n^2 $(2n^3)$ operations. The solution of a $(n \times n+1)$ system of linear equations is assumed to require $\simeq (n^3+3n^2)$ operations.

TABLE 2.2

	N	M	K	R	$\alpha = 0.9$		$\alpha = 0.8$		$\alpha = 0.6$	
					Time (sec)	Iterations [a]	Time (sec)	Iterations [a]	Time (sec)	Iterations [a]
(Monotone)										
1.....	3	5	15	1	4.13	4	4.10	4	3.15	3
				5	5.87	3	5.90	3	6.07	3
				10	9.53	3	9.50	3	9.55	3
2.....	3	10	7	1	5.43	3	5.38	3	7.02	4
				5	9.82	3	10.05	3	9.68	3
				10	15.26	3	15.30	3	14.96	3
3.....	3	15	5	1	10.23	4	10.38	4	7.58	3
				5	13.95	3	13.83	3	14.03	3
				10	21.08	3	21.63	3	21.35	3
4.....	5	5	10	1	8.63	5	6.80	4	6.88	4
				5	9.78	3	5.70	2	9.78	3
				10	15.42	3	8.68	2	8.55	2
5.....	5	10	5	1	12.23	4	9.08	3	9.22	3
				5	16.55	3	16.45	3	9.52	2
				10	25.92	3	14.25	2	14.55	2
(Single favourable)										
6.....	3	5	15	1	5.20	5	5.03	5	3.05	3
				5	8.50	4	5.75	3	3.47	2
				10	13.78	4	9.37	3	5.25	2
7.....	3	10	7	1	7.20	4	5.37	3	5.21	3
				5	9.80	3	9.70	3	9.38	3
				10	14.90	3	15.01	3	15.16	3
8.....	3	15	5	1	14.90	6	12.77	5	9.68	4
				5	19.10	4	13.78	3	13.28	3
				10	21.00	3	20.96	3	11.87	2
9.....	5	5	10	1	8.20	5	4.90	3	5.37	3
				5	13.22	4	9.87	3	9.93	3
				10	14.83	3	15.92	3	15.75	3
10.....	5	10	5	1	18.43	6	15.25	5	12.08	4
				5	16.52	3	16.37	3	16.30	3
				10	24.85	3	25.95	3	14.32	2

[a] The final iteration includes only a single policy improvement step (i.e., $R = 1$), indicating final policy cannot be improved. At a tolerance of one part in 10^{13}, the value iteration termination criteria was totally ineffective.

was always set to the same arbitrary policy and the algorithm was initialized with a policy evaluation step. Over 20 other problems were solved with similar results. The programs which were written in APL and run on an IBM 360/91 are not indicative of the minimum running times achievable, but are sufficient to indicate the number of iterations one might expect and the relative effectiveness of performing an increased number (R) of policy improvement iterations for each policy evaluation.

Notice that in essentially all the problems above, the use of multiple policy improvement reduced the number of iterations required, but the running time was reduced only in problems $4(\alpha = 0.8)$ and $10(\alpha = 0.9)$. Also, notice that the number of iterations required is consistently of the order of N, the number of states in a subgroup, and decreases as the discount factor decreases. These results indicate an advantage of unmodified policy iteration (i.e., $R = 1$) over value iteration approaches (i.e., $R > 1$), and suggest that the number of required iterations will grow only linearly with the number of states in a subgroup, rather than with the total number of states in the problem.

REFERENCES

[1] Blackwell, D., "Discrete Dynamic Programming," Ann. Math. Stat. *33*, 719–726 (1962).

[2] De Ghellinck, G. T. and G. D. Eppen, "Linear Programming Solutions for Separable Markovian. Decision Problems," Management Science *13*, 371–394 (Jan. 1967).

[3] Denardo, E. V., "Contraction Mappings in the Theory Underlying Dynamic Programming," SIAM Review *9*, 165–177 (Apr. 1967).

[4] Denardo, E. V., "Separable Markovian Decision Problems," Mangement Science *14*, 451–462 (Mar. 1968).

[5] D'Epeneux, F., "A Probabilistic Production and Inventory Problem," Management Science *10*, 98–109 (Oct. 1963).

[6] Hartman, J. K. and L. S. Lasdon, "A Generalized Upper Bounding Method for Doubly Coupled Linear Programs," Nav. Res. Log. Quart. *17*, 411–429 (Dec. 1970).

[7] Howard, R. A., *Dynamic Programming and Markov Processes* (Technological Press of M.I.T., Cambridge, 1960).

[8] Kalymon, B. A., "Stochastic Costs in Multi-Period Decision Models," Ph.D. Dissertation, Yale University (1970).

[9] Kalymon, B. A., "Stochastic Prices in a Single Item Inventory Purchasing Model," Operations Research *19*, 1434–1458 (Oct. 1971).

[10] Kalymon, B. A., "Machine Replacement with Stochastic Costs," Management Science *18*, 288–298 (Jan. 1972).

[11] Kaufman, A. and R. Cruon, *Dynamic Programming-Sequential Scientific Management* (Academic Press, New York, 1967).

[12] Kemeny, J. G. and J. L. Snell, *Finite Markov Chains* (Van Nostrand, Princeton, N.J., 1960).

[13] Ross, S., *Applied Probability Models with Optimization Applications*, (Holden-Day, San Francisco, Calif., 1970).

POLAROIDS: A NEW TOOL IN NON-CONVEX AND IN INTEGER PROGRAMMING

Claude-Alain Burdet

Graduate School of Industrial Administration
Carnegie-Mellon University

ABSTRACT

This paper presents a generalization, called *polaroid*, of the concept of *polar* sets. A list of properties satisfied by polaroids is established indicating that the new concept may be fruitfully used in an area of non-convex (called here polar) programming as well as in integer programming, by means of polaroid cuts; this class of new cuts contains the ones defined by Tuy for concave programming (a special case of polar programming) and by Balas in integer programming; it furthermore provides for new degrees of freedom in the construction of algorithms in the above-mentioned areas of mathematical programming.

1. PRELIMINARIES

In this introductory section we present the definitions and some relevant properties of a new concept: polaroid sets and functions; these mathematical objects are derived directly from the theory of convex sets and represent a generalization of polar sets; historically, polar relationships have been one of the main topics in projective geometry where the involutory correspondence of poles and polars is of prime interest. The first generalization to polar sets can be found in Minkowski [10]; these sets have since played an increasingly important role in the area of convex analysis and methematical programming (see for instance, the treatise [11] by Rockafellar); more recently, Balas [2] has uncovered an interesting application of polar sets to integer programming; there the use of a positive-definite quadratic form (n-dimensional sphere or ellipsoid) allows him to define his outer-polar sets which enjoy all the desirable properties for the convex outer-domain theory of valid intersection cuts (further aspects of this theory have been discussed by Glover [7] and Burdet [3, 4]).

Our objective here is merely to present some fundamental properties of polaroids, and to indicate how they can be fruitful in an area of nonconvex programming (called here polar programming) as well as in integer (primarily zero-one) programming.

When the constrained set is polyhedral, the use of polaroids can be viewed as a generalization of the intersection approach of Hoang Tuy [12] for concave programming or of the intersection cut approach initiated by Balas [1] in integer programming. The generalization with respect to [12] is that polaroid cuts can be defined for nonconcave problems; furthermore, for concave problems Tuy's cuts are uniformly dominated by polaroid cuts; indeed Tuy's approach can be imbedded as a trivial special case in the present theory. With respect to the results of Balas [1, 2], the generalization consists in the fact that:

(a) convex polaroids can be defined for a more general class of functions than positive definite quadratic forms as in [2]; adequately constructed polaroids can be used as convex outer-domains to generate new cutting planes.

(b) the convexity requirement imposed on outer-domains merely plays the role of a sufficiency condition in the construction of an intersection cut, and it can be relaxed; it is shown that nonconvex polaroids may very well be used to generate valid polaroid cuts.

MEA CULPA: The underlying goal of this report is not primarily focused on numerical and computational aspects of the uncovered properties; the basic idea was to start with as general an object as possible (viz. polaroid sets) and to test those properties which seem promising for the global optimization of nonconvex problems. The analysis resulted in a hierarchy of properties ranked by increasingly strong assumptions. In a second effort it was found, however, that many of the additional assumptions leading to the more sophisticated (higher ranked) properties (such as convexity) certainly are convenient because they provide for automatic sufficiency conditions, but at the same time they seem to be unnecessarily taming the power of the approach. For instance, for the construction of a cutting plane it is believed (as indicated in the conclusions) that a fruitful line of research (in order to attain computational efficiency) would be to proceed along the following way:

(a) generate a cut under very weak a priori assumptions (from an arbitrary polaroid, for instance).

(b) check a posteriori the validity of the cut (using Theorems 13 or 14); this may require an ad hoc weakening readjustment of the cutting plane, but will, in general, yield a better cut than if sufficiency conditions had been postulated a priori for the entire polaroid.

2. POLAROIDS

Let $f=f(x, y)$† be a real valued function with two n-vector arguments x and y; let P denote a closed set in $\mathscr{R}^n$.

DEFINITION 1: For a given value of the parameter k, define the polaroid set $P^*(k)$ by

$$P^*(k) = \{y \mid f(x, y) \leqslant k, \forall x \epsilon P\}. \tag{1}$$

By convention set $P^*(k) = \phi$, $\forall k < \min_y \{f(x, y) \mid x \epsilon P\}$.

THEOREM 1 (*Inclusion theorem*): For any closed sets P and $Q \subset \mathscr{R}^n$, one has the following implications:

If $Q \subset P$, then $Q^*(k) \supset P^*(k)$; and

If $k_1 \geqslant k_2$, then $P^*(k_1) \supset P^*(k_2)$.

PROOF: The assertions follow from (1):

$$Q^*(k) = \{y \mid f(x, y) \leqslant k, \forall x \epsilon Q\} \supset \{y \mid f(x, y) \leqslant k, \forall x \epsilon P \supset Q\} = P^*(k)$$

$$P^*(k_1) = \{y \mid f(x, y) \leqslant k_1, \forall x \epsilon P\} \supset \{y \mid f(x, y) \leqslant k_2 \leqslant k_1, \forall x \epsilon P\} = P^*(k_2) \qquad \text{Q.E.D.}$$

THEOREM 2 (*Union theorem*): For any closed sets P, Q one has

$$(P \cup Q)^*(k) = P^*(k) \cap Q^*(k).$$

---

†f is, in fact, a bifunction [11] with object functions $f(x; \cdot)$, for $x \epsilon P$; f can also be viewed as a multivalued mapping $\mathscr{R}^n \to \mathscr{R}$, where each $x \epsilon P$ determines a particular mapping.

PROOF:

$$(P \cup Q)^*(k) = \{y \mid f(x, y) \leq k, \forall x \in (P \cup Q)\}$$

$$= \{y \mid f(x, y) \leq k, \forall x \in P\} \cap \{y \mid f(x, y) \leq k, \forall x \in Q\}$$

$$= P^*(k) \cap Q^*(k). \qquad \text{Q.E.D.}$$

COROLLARY 2.1: Let

$$(P \Delta Q) = cl\{x \mid x \in (P \cup Q) \text{ and } x \notin (P \cap Q)\}.$$

Then one has

$$(P \cap Q)^*(k) \cap (P \Delta Q)^*(k) = P^*(k) \cap Q^*(k).$$

PROOF:

$$(P \cup Q)^*(k) = [(P \cup Q) \cup (P \Delta Q)]^*(k) = (P \cap Q)^*(k) \cap (P \Delta Q)^*(k)$$

On the other hand, Theorem 2 implies:

$$(P \cup Q)^*(k) = P^*(k) \cap Q^*(k) \qquad \text{Q.E.D.}$$

THEOREM 3 (*Intersection theorem*):

$$(P \cap Q)^*(k) \supset P^*(k) \cup Q^*(k).$$

PROOF:

$$(P \cap Q)^*(k) = \{y \mid f(x, y) \leq k, \forall x \in (P \cap Q)\} \supset \{y \mid f(x, y) \leq k, \forall x \in P\} \cup \{y \mid f(x, y) \leq k, \forall x \in Q\}$$

$$= P^*(k) \cup Q^*(k). \qquad \text{Q.E.D.}$$

COROLLARY 2.2:

$$P^*(k) = (P \cap Q)^*(k) \cap (P \Delta (P \cap Q))^*(k)$$

$$Q^*(k) = (P \cap Q)^*(k) \cap (Q \Delta (P \cap Q))^*(k).$$

PROOF: $P = (P \cap Q) \cup (P \Delta (P \cap Q))$, and similarly for Q; applying the union theorem completes the proof. Q.E.D.

COROLLARY 2.3:

$$P^*(k) \cup Q^*(k) = (P \cap Q)^*(k) \cap [(P\Delta(P \cap Q))^*(k) \cup (Q\Delta(P \cap Q))^*(k)]$$

$$\supset (P \cap Q)^*(k) \cap (P\Delta Q)^*(k) = P^*(k) \cap Q^*(k).$$

PROOF: The (first) equality is obtained directly from the corollary 2.2; the inclusion can be derived by inspection of the polaroid sets in the bracket or directly from Corollary 2.1, since $P^* \cup Q^* \supset P^* \cap Q^*$.

DEFINITION 2: The function $g = g(x) = f(x, x)$ is said polarized by f. Denote the level set of g by $\text{lev}_\alpha\, g = \{x \mid g(x) \leq \alpha\}$.

DEFINITION 3: An arbitrary (closed compact) set S satisfying

$$\phi \neq S \cap P \subset \text{lev}_k\, g$$

is called a valid cut at the level k.

THEOREM 4: A cut S, valid at the level k_1, is valid at all higher levels $k_2 (\geq k_1)$.

PROOF: Immediate since $\text{lev}_{k_1} g \subset \text{lev}_{k_2} g$ by definition. Q.E.D.

3. CONVEX POLAROIDS

THEOREM 5: The polaroid $P^*(k)$ is convex $\forall k$ if $f(x, y)$ is quasi-convex in y, for all $x \epsilon P$.

PROOF: Let y^1 and y^2 be arbitrary points in $P^*(k)$; and set

$$y^3 = \lambda y^1 + (1 - \lambda) y^2, \; 0 \leq \lambda \leq 1,$$

then $\forall x \epsilon P$, one has

$$f(x, y^3) = f(x, \lambda y^1 + (1 - \lambda) y^2) \leq \max \{f(x, y^1), f(x, y^2)\} = k. \quad \text{Q.E.D.}$$

In order to acquire, at this point, a better geometrical feeling for the content of the statement in Theorem 5 let us review some classical results:

(1) Let f be the euclidean scalar product

$$f(x, y) = \sum_{i=1}^{n} x_i y_i = < x, y >$$

which yields (the square of) the euclidean norm as polarized function

$$g(x) = \sum_{i=1}^{n} x_i^2.$$

Following the classical definition [11] one finds that, in this case, the polaroid $P^*(1)$ is nothing, but the polar set P^*

$$P^*(1) = P^* = \{x^* \mid < x, x^* > \leq 1, \forall x \epsilon P\}.$$

In [2] Balas introduced a generalization of the classical concept by allowing for a scaling factor, k, in the above definition. Clearly this can be absorbed in our definition, either by considering the polaroid, $P^*(k)$, with respect to the same polaroid function f, or by changing the polaroid function to $\tilde{f} = \frac{1}{k} f$ and retaining the parameter, $1 : \tilde{P}^*$ (1).

Since f is bilinear it is quasi-convex in y and Theorem 5 paraphrases, in this case, the convex property of polar sets.

(2) Let f be the scalar product function corresponding to a general Riemann metric, with arbitrary (real symmetric) metric tensor g_{ik}; i.e.,

$$f(x, y) = \sum_{i,k=1}^{n} g_{ik} x_i y_k,$$

for the polarized function, g, one has

$$g(x) = \sum_{i,k=1}^{n} g_{ik} x_i x_k.$$

Since the parameter k, can always be absorbed by the tensor g_{ik}, the polaroid $P^* = P^*(1)$ is a genuine generalization of the polar concept, to arbitrary quadratic forms (not necessarily definite or positive).

Since f is again bilinear, one finds from Theorem 5 that the polaroid P^*, generated from an arbitrary set P is convex; in a follow-up paper, we show how the polaroid theory can be used to solve an arbitrary nonconvex quadratic programming problem (particularly in the indefinite case) [5].

(3) a. Let $F(x)$ be a vector valued function and define $f(x, y)$ to be the scalar product

$$f(x, y) = F(x)y.$$

b. Let $g(x)$ be differentiable and define

$$f(x, y) = g(x) + (y - x) \forall g(x).$$

c. Let $F = F(x)$ and $l = l(y)$ be real valued, and l a quasi-convex function; then set

$$f(x, y) = F(x) l(y).$$

(4) Consider a quasi-convex function $g(x)$ and define

$$f(x, y) = g(y), \forall x.$$

The polaroid $P^*(k)$ then merely reduces to $P^*(k) = \text{lev}_k g$ and Theorem 5 restates the known convexity property of $\text{lev}_k g$. In general, however, there belongs many other possiblities for choosing $f = f(x, y)$,

such that $f(x, x) = g(x)$; hence there will correspond other polaroid sets to the same polarized function $g(x)$; Example 1 illustrates this situation, for instance.

(5) Let $f(x, y)$ satisfy $\forall x \in P$:

a. $f(x, y) \geq 0$

b. $f(x, \lambda y) = \lambda f(x, y)$, $\lambda \geq 0$.

Then $P^*(k)$ contains the origin $y = 0$ and it is star-shaped.

PROOF: Let $y \in P^*(k)$ i.e., $f(x, y) \leq k$, $\forall x \in P$; then $\lambda y \in P^*(k)$, for $k \geq 0$, $\forall \lambda \in [0, 1]$ because $0 \leq f(x, \lambda y) = \lambda f(x, y) \leq \lambda k \leq k$. Q.E.D.

4. COMPLETE POLAROIDS

Until now we always considered k as an accessory parameter whose value played no essential role. We now analyze polaroids (convex or not) corresponding to particular values of k.

DEFINITION 4: The polaroid $P^*(k)$ defined by (1) is called *complete* if

$$P \subset P^*(k).$$

DEFINITION 5: A function $f = f(x, y)$ is called complete on P at the level k if

$$f(x, y) \leq k, \forall x, y \in P.$$

THEOREM 6 (*Completeness Theorem*): The polaroid $P^*(k)$ is complete iff f is complete on P (at the level k).

PROOF: Take any $x' \in P$; then $f(x, x') \leq k$, $\forall x \in P$ by hypothesis, when f is complete; but one has $P^*(k) = \{y | (f(x, y)) \leq k, \forall x \in P\}$, which shows that x' must belong to $P^*(k)$ and hence $P \subset P^*(k)$. Conversely, suppose $P^*(k)$ complete; the argument is by contradiction: suppose there exists a pair x, $y \in P$ such that $f(x, y) > k$; in this case one has either $x \notin P^*(k)$ or $y \notin P^*(k)$, or both; in any case $P \not\subset P^*(k)$ which is contrary to hypothesis. Q.E.D.

The following theorems will be useful in establishing optimality conditions for polar programming as well as for testing the validity of polaroid cuts in integer programming.

COROLLARY 6.1: If $P^*(k)$ is complete then $P \subset lev_k g$.

PROOF: By hypothesis one has $f(x, y) \leq k$, $\forall x$, $y \in P$; hence, in particular for $y = x$: $f(x, x) = g(x) \leq k$. Q.E.D.

THEOREM 7: If f is symmetric, i.e., $f(x, y) = f(y, x)$ then

$$P \subset (P^*(k))^*(k).$$

PROOF: $P^*(k) = \{y | f(x, y) \leq k, \forall x \in P\}$, hence $f(x, y) \leq k$, $\forall x \in P$, $V y \in P^*(k)$; but by hypothesis $f(x, y) = f(y, x) \leq k$, hence $\forall x \in P$, one has $x \in (P^*(k))^*(k) = \{y | f(y, u) \leq k, \forall y \in P^*(k)\}$, and therefore $P \subset (P^*(k)^*(k)$. Q.E.D.

COROLLARY 7.1: If $P^*(k)$ is complete (and f symmetric) then

$$P \subset (P^*(k))^*(k) \subset P^*(k).$$

PROOF: $P \subset P^*(k)$ implies

$$(P^*(k))^*(k) = \{y | f(x,y) \leq k, \forall x \in P^*(k)\} \subset \{y | f(x,y) \leq k, \forall x \in P\} = P^*(k). \qquad \text{Q.E.D.}$$

THEOREM 8: $P \supset (P \cap \text{lev}_k\, g) \supset (P^*(k) \cap P)$.

PROOF: $(P \cap \text{lev}_k\, g) = \{x \,|\, g(x) \leq k, x \in P\}$

$$\supset \{y \in P \,|\, f(x,y) \leq k, \forall x \in P\} = (P^*(k) \cap P). \qquad \text{Q.E.D.}$$

Let us now consider the "boundary" sets†

$$bdP^*(k) = \{y \in P^*(k) | f(x,y) = k, \quad \text{for some } x \in P\}$$

$$bd(\text{lev}_k\, g) = \{x | g(x) = k\}.$$

COROLLARY 8.1: Suppose $P^*(k)$ is complete, then one has

$$[P \cap bd(\text{lev}_k\, g)] = [P \cap bdP^*(k)].$$

PROOF: Since by definition "$bd\ Q$" is a subset of Q, Theorem 8 provides one inclusion; the inclusion in the reverse direction is now established: take $x \in [P \cap$ "bd $(\text{lev}_k\, g)$"], i.e., $x \in P$ with $g(x) = k$. Since $P^*(k)$ is complete, one also has $x \in P^*(k)$. Thus, $x \in P^*(k)$ and $g(x) = f(x,x) = k$, with $x \in P$ which completes the proof.‡ Q.E.D.

Theorem 6 indicates that when $P^*(k)$ is complete, k is an upper bound for the polarized function, g, on the set P. Theorem 8 now states that this upper bound may only be attained on the "boundary" of the polaroid $P^*(k)$. Thus completeness means that no interior point of $P^*(\bar{k})$ is an optimal polar program $\bar{x} \in P$, with $\bar{k} = \max_{x \in P} g(x) = g(\bar{x})$. This is stated in the following:

COROLLARY 8.2 (*Boundary Theorem*): If $P^*(\bar{k})$ is complete, then every optimal solution $\bar{x}$ of the polar programming problem

$$\text{maximize } g(x), \qquad \text{subject to } x \in P$$

satisfies $\bar{x} \in bdP^*(\bar{k})$.

PROOF: Optimality implies $\bar{x} \in bd$ $\text{lev}_k\, g$; Corollary 8.1 completes the proof. Q.E.D.

The above corollary, in a way, corresponds to the classical result asserting that the maximum of a quasi-convex function over a closed (bounded) set P is always attained on the boundary of P.

COROLLARY 8.3: (Sufficient Optimality Theorem)

The point $\bar{x} \in P$ an optimal solution of the *polar programming* problem

$$\text{maximize } g(x), \qquad \text{subject to } x \in P$$

†The term "boundary" has been adopted here because this definition corresponds to the intuitive concept for the important special case where $P^*(k)$ is convex; for an arbitrary function f however, P^* need not be convex and "bd" P^* no longer need correspond to the topological concept.

‡A more general result, which does not require completeness of $P^*(k)$ is established in [5].

if the polaroid $P^*(\bar{k})$ with $\bar{k}=g(\bar{x})$ is complete, i.e., $P \subset P^*(\bar{k})$.

PROOF: By hypothesis the point $\bar{x}$ lies on the "boundary" set: $\bar{x} \in$ "bd $\text{lev}_{\bar{k}} g$"; furthermore $P^*(\bar{k})$ is assumed complete, hence (Corollary 6.1) $P \subset \text{lev}_{\bar{k}} g$, implying that $\bar{k}$ is an upper bound. But, since $\bar{k}=g(\bar{x})$, $\bar{x} \in P$ is optimal. Q.E.D.

COROLLARY 8.4: (Necessary Optimality Theorem)

Optimality of the solution $\bar{x}$, $\bar{k}=g(\bar{x})$ of the polar programming problem implies

a. $\bar{x} \in$ "Int $P^*(k)$", $\forall k > \bar{k}$.

b. $\bar{x} \notin P^*(k)$, $\forall k < \bar{k}$.

c. $\bar{x} \in$ "bd $P^*(\bar{k})$".

PROOF: First note that c. implies $\bar{x} \in P^*(k)$, $\forall k \geq \bar{k}$ (Theorem 1) and that $\bar{k}=g(\bar{x})=f(\bar{x}, \bar{x})$ implies c. (Corollary 8.1).

Furthermore, one has

$$(P \cap \text{lev}_k g) \supset (P \cap P^*(k))$$

hence, since $\bar{x} \notin \text{lev}_k g$, $\forall k < \bar{k}$ and $\bar{x} \in P$, b holds true. Moreover, since $\bar{x} \in \text{Int}\ (\text{lev}_k g)$, $\forall k > \bar{k}$ one has a. Q.E.D.

The sufficiency condition of Corollary 8.3 is useful in concave programming; for the case where $P^*(k)$ is not complete, a necessary and sufficient optimality condition is established in [5].

THEOREM 9: Let $Q \subset P$; if $P^*(k)$ is complete, then $Q^*(k)$ is complete.

PROOF: From the inclusion Theorem 1 one has $P^*(k) \subset Q^*(k)$ and completeness of $P^*(k)$ yields

$$Q \subset P \subset P^*(k) \subset Q^*(k).$$ Q.E.D.

THEOREM 10: Let $Q \subset P$ and $P^*(k)$ be complete; then one has

and

$$Q \subset P \subset \text{lev}_k g$$

$$(\text{lev}_k g)^*\ (k) \subset P^*(k) \subset Q^*(k).$$

PROOF: Immediate from Theorems 1 and 7. Q.E.D.

COROLLARY 10.1: If $P^*(k)$ and $(\text{lev}_k g)^*\ (k)$ are complete, then one has

$$Q \subset P \subset \text{lev}_k g \subset (\text{lev}_k g)^*\ (k) \subset P^*(k) \subset Q^*(k).$$

PROOF: Immediate from Theorem 10. Q.E.D.

THEOREM 11: If $(P \cup Q)^*\ (k)$ is complete, then both $P^*(k)$ and $Q^*(k)$ are complete.

PROOF: From the union theorem one has

$$(P \cup Q) \subset (P \cup Q)^*\ (k) = P^*(k) \cap Q^*(k);$$

but

$$P \subset (P \cup Q) \qquad \text{so that} \qquad P \subset P^*(k) \cap Q^*(k);$$

and hence $P \subset P^*(k)$; and similarly for Q. Q.E.D.

In integer programming, one is really interested in the polaroid set of isolated sets (points in all interger-, or linear fibers in mixed integer-programming). Completeness of the outer-domain is clearly a desired feature in order to generate a deep cut and Theorem 11 indicates that such an outer domain can be constructed as the intersection of individual polaroids (one for each point or fiber) provided all such polaroids contain all feasible integer points.

The same argument holds true, of course, in polar programming if the feasible set P consists of (or is arbitrarily split into) several components.

5. VALIDITY AND OPTIMALITY CONDITIONS

Definition 3 of a valid cut allows one to consider optimality conditions in polar programming as a particular valid cut; this is formulated in the next

THEOREM 12: The polarized function $g(x)=f(x, x)$ attains its maximal value $\bar{k}=g(\bar{x})$ over the set P at the point $\bar{x}\epsilon P$ if there exists a valid cut S at the level $\bar{k}=\max_{x\epsilon P} g(x)=g(\bar{x})$.

PROOF: Trivial (take for instance $S=\text{lev}_{\bar{k}}\, g$) Q.E.D.

In this section we are not only interested in stating necessary and sufficient optimality conditions for polar programs in terms of more general (and computationally more easily tractable) polaroids than the (trivial) level sets lev g· we also want to find the conditions which must be satisfied, for an arbitrary cut to be valid; for instance, the term valid cut can here be visualized as stemming from the cutting plane approach of Tuy [12] for concave programming, or from the intersection cut approach, in integer programming [1, 4, 7]. In the latter case, however, a somewhat stronger concept for valid cuts is necessary (see section 7).

THEOREM 13: If the set $(P\cap S)^*(k)$ is complete then S is a valid cut at the level k.

PROOF: Immediate from Theorem 7, applied to the set $(P\cap S)$.

THEOREM 14: Every subset S of a polaroid $P^*(k)$ is a valid cut.

PROOF: (Eliminate the uninteresting case where $S\cap P=\phi$.) One has

$$P^*(k)=\{y|f(x,y)\leq k, \forall x\epsilon P\},$$

hence $f(x,y)\leq k, \forall x\epsilon P, \forall y\epsilon S\subset P^*(k)$, and, in particular,

$$f(x,y)\leq k, \forall x, y\epsilon P\cap S.$$

Thus the function, f, is complete on the set $(P\cap S)$; Theorems 6 and 7 establish that Definition 3 is satisfied. Q.E.D.

COROLLARY 14.1: $P\cap P^*(k)\subset \text{lev}_k\, g$.

PROOF: (1) One may simply set $S=P^*(k)$ and apply Theorem 14, together with Definition 3.

(2) Alternately, one has from Theorem 8 $(P^*(k)\cap P)\subset (P\cap \text{lev}_k\, g)\subset \text{lev}_k\, g$. Q.E.D.

In Theorems 13 and 14 one notes that neither of them requires completeness of $P^*(k)$; all that is required in Theorem 13 is that the cutoff portion of P (i.e., $(P\cap S)$) be contained in the polaroid $(P\cap S)^*(k)$; from the intersection Theorem 3 one has

$$(P\cap S)^*(k)\supset P^*(k)\cup S^*(k),$$

which shows that the hypothesis of Theorem 13 is quite weak and should be easy to test computationally; in particular, when one constructs S as a subset of $P^*(k)$ (as is often practically the case) Theorem 14 shows that one merely must ensure $S \subset P^*$.

6. POLAROID CUTS FOR LINEARLY CONSTRAINED POLAR OR INTEGER PROGRAMMING PROBLEMS

Let us now focus our attention on the polyhedral sets $P \subset \mathscr{R}^n$, i.e.,

$$P=\{x=x_1, \ldots, x_n) \mid x_i=\bar{x}_i-\sum_{j\in\bar{N}} \bar{a}_{ij}t_j \geq 0, \forall i\in(N \cup M)\}, \tag{2}$$

where it is assumed that (2) represents a linear program in explicit (Tucker) format, expressed with respect to a basis with nonbasic set $\bar{N} \subset (N \cup M)$; $N=\{1, 2, \ldots, n\}$ is the set of the original variables; $M=\{n+1, \ldots, n+m\}$ is the set of the slack (and artificial) variables. Consider the extreme (basic) ray $u^j(t_j)=\bar{x}-\bar{a}_j t_j$, $t_j \geq 0$, $j\in\bar{N}$, where

$$\bar{a}_j=(\bar{a}_{1j}, \bar{a}_{2j}, \ldots, \bar{a}_{nj})$$

and

$$\bar{x}=(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n),$$

and assume (for simplicity of the exposition) that $\bar{x}\in\text{Int } P^*(k)$, i.e., that the intersection points $u^j(\bar{\lambda}^j)$ of each ray $u^j, j\in\bar{N}$ with the "boundary" of $P^*(k)$ are different from $\bar{x}$, that is

$$u^j(\bar{\lambda}^j)=\bar{x}-\bar{\lambda}^j\bar{a}_j, \qquad \text{with } \bar{\lambda}^j>0.$$

Furthermore define the halfspaces

$$H^+=\left\{x=x(t) \;\middle|\; \sum_{j\in\bar{N}} \frac{1}{\bar{\lambda}^j} t_j > 1\right\}$$

$$H^-=\left\{x=x(t) \;\middle|\; \sum_{j\in\bar{N}} \frac{1}{\bar{\lambda}^j} t_j < 1\right\},$$

where $x(t)=(x_1(t), x_2(t), \ldots, x_n(t))$ is given by the n first components of the linear program (2) characterizing P. Clearly H^+ and H^- are open, and they are both defined by the cutting plane

$$\sum_{j\in\bar{N}} \frac{1}{\bar{\lambda}^j} t_j = 1.$$

(Note that, by assumption $\bar{x} \in H^-$, i.e., $\bar{x} \notin H^+$.)

Define the set $S \subset P^*(k)$ as the following n-dimensional simplicial hull

$$S = \text{conv } \{\bar{x}, u^j(\bar{\lambda}^j), \forall j\in\bar{N}\}.$$

Assuming that the hypothesis of either Theorem 13 or 14 is satisfied, the set S represents a valid cut (see section 7). Implementation into the current linear programming tableau merely amounts to the addition of the new linear constraint,

$$\sum_{j\in\bar{N}} \frac{1}{\lambda^j} t_j \geq 1,$$

called polaroid cut.

For the use of a cutting plane method to solve nonconvex problems the reader is referred to the papers by Hoang Tuy [12], Glover [7] and Klingman [8], Gomory [9], Balas [1, 2] or Burdet [3, 6]. A more detailed study of particular types of polaroid cuts in integer programming is given in Burdet [4, 5].

7. CONCLUSIONS

In polar programming the objective function is the polarized function g; algorithms for solving such nonconvex optimization problems therefore follow a stepwise construction of a set S satisfying

$$P \subset S \subset P^*(\bar{k}),$$

where $\bar{k}$ is always the current best value and ultimately $\bar{k} = \max_{x\in P} g(x)$. The optimality theorem provides for the stopping criterion. In practice, the use of polariods will prove efficient whenever the set $P^*(\bar{k})$ is much larger than P; in this case ample room is left for an easy construction of a set S yielding a sufficiency condition for global optimality.

In integer programming, the use of polaroids is somewhat different since they are only used for the characterization of the set of feasible integer solutions, as compared to the other (continuous) feasible solutions; the polarized function g then usually has the property:

$g(v) = 1$ $\quad \forall\, v =$ vertex of a unit cube $U(\bar{x})$ containing the linear programming optimum $\bar{x}$;

$g(x) \leq 1$ $\quad \forall x \in U(\bar{x})$;

thus $\bar{k} = 1$; the meaning of a valid cut S here becomes: Int(S) contains no feasible integer point (as shown by Theorem 13, the construction of a valid cut requires neither completeness nor convexity of the polaroid, P^*). However, a stronger definition of valid cuts is needed here; we need add the requirement

$$\text{Int}(S) \cap [P \cap bd\ lev_k g] = \phi,$$

to Definition 3.

In terms of this new definition, Theorem 13 remains true if one requires the additional hypothesis: $S \subset (P \cap S)^*(k)$; this implies that every feasible integer point which lies in S, lies on the boundary of S:

PROOF: From Theorem 8 one has

$$\text{Int}\,(S) \cap ((P \cap S) \cap (P \cap S)^*(k)) \subset \text{Int}\,(S) \cap ((P \cap S) \cap \text{lev}_k\, g)$$

but the latter is the empty set.

This general result opens a new area of research which contrasts from the algebraic approach of Gomory [9] or the intersection approach of Balas [1] where one imposes in advance conditions which are sufficient to ensure the validity of the cut independently of the particular simplex S which is actually generated. The polaroid cut approach also entails such a possibility (using convex and complete polaroids); but, in addition, it can lead to new methods where an arbitrary polaroid P^*, is used to generate the simplex S (owing to the generality of P^*, this cut can be made deep); in a second phase, the validity of S is then established; for instance by checking directly $S \subset P^*$ (Theorem 14).

REFERENCES

[1] Balas, E., "Intersection Cuts—A New Type of Cutting Planes for Integer Programming." Operations Research, Vol. 19, No. 1, Jan.–Feb. 1971, pp. 19–39.

[2] Balas, E., "Integer Programming and Convex Analysis." Mathematical Programming, Vol. 2, No. 3 (1972), pp. 330–381.

[3] Burdet, C. A., "Enumerative Inequalities in Integer Programming." Mathematical Programming, Vol. 2, No. 1 (1972), pp. 32–64.

[4] Burdet, C. A., "Enumerative Cuts I and II." To appear, Operations Research (1973).

[5] Burdet, C. A., "Simple Polaroids for Non-Convex Quadratic Programming," Carnegie-Mellon University W.P. 101–71–2 (May, 1972).

[6] Burdet, C. A., "The Facial Decomposition Method," Carnegie-Mellon University W.P. 104–71–2 (May, 1972).

[7] Glover, F., "Cut-Search Methods in Integer Programming." University of Colorado, September 1970.

[8] Glover, F. and D. Klingman, "Concave Programming applied to a special class of 0–1 Integer programs." University of Texas at Austin (1971).

[9] Gomory, R., "On the Relation Between Integer and Noninteger Solutions to Linear Programs." Proceedings of the National Academy of Sciences *53* (1965), pp. 260–265.

[10] Minkowski, H., "Theorie der Konvexen Körper insbesondere Begründung ihres Oberflächenbegriffes." Gesammelte Abhandlungen, Vol. 2, Leipzig (1911).

[11] Rockafellar, R. T., *Convex Analysis*. Princeton University Press, 1970.

[12] Tuy, Hoang, "Concave Programming Under Linear Constraints" (Russian). Doklady Akademii Nauk SSSR, 1964. English translation in Soviet Mathematics, 1964, pp. 1437–1440.

[13] Tuy, Hoang, "Vê môt lop qui hoach phi-tuyên," Toán Lý, No. 1, 1963. Publications de la Section de Physique et Mathématiques du comité d'Etat des Sciences, République Démocratique du Vietnam.

A SQUARED-VARIABLE TRANSFORMATION APPROACH TO NONLINEAR PROGRAMMING OPTIMALITY CONDITIONS*

James G. Taylor†

Naval Postgraduate School
Monterey, California

ABSTRACT

We show that the well-known necessary and sufficient conditions for a relative maximum of a nonlinear differentiable objective function with nonnegative variables constrained by nonlinear differentiable inequalities may be derived using the classical theory of equality-constrained optimization problems with unrestricted variables. To do this we transform the original inequality-constrained problem to an equivalent equality-constrained problem by means of a well-known squared-variable transformation. Our major result is to show that second order conditions must be used to obtain the Kuhn-Tucker conditions by this approach. Our nonlinear programming results are motivated by the development of some well-known linear programming results by this approach.

1. INTRODUCTION

We do not claim to derive any new results in this paper, and we do not claim to use any new methods. However, we do present an elementary derivation of the Kuhn-Tucker [32] conditions as an extension of well-known results from classical equality constrained optimization theory (see [1]). We show that the well-known "squared-variable transformation" approach to mathematical programming problems of creating an equivalent equality constrained problem with unrestricted variables only yields the Kuhn-Tucker conditions when second order [27], [34] conditions are also considered. This apparently has never been noted in the literature previously, although (as Dantzig noted in 1956 [18]) most people who have considered problems involving inequalities have been over this ground. The crucial point is, however, that they all have stopped just short of our present development.

Since the appearance of the papers by John [28] and Kuhn and Tucker [32], there have appeared a number of alternate derivations of necessary (and sometimes sufficient) conditions of optimality for inequality constrained optimization problems by Dorn [20], Wilde [41], [42], Bernholtz [7], Hadley (see pp. 190–194 in [26]), King [29], and others. These particular papers have developed their results by using more "elementary" arguments, not involving Farkas' Lemma [21] of the theory of linear inequalities (see also [28], [35], [36]). Besides their generalization of the notion of Lagrange multipliers (see also [39]) to constrained optimization problems with inequality constraints, Kuhn and Tucker also extensively discussed the saddle-point property of Lagrangian functions (see also [36]), first observed by Courant and Hilbert (see pp. 231–233 of [16]) and used by the latter as motivation for the concept of equivalent variational problems in the calculus of variations (as developed by C. Carathéodory (see [13])).

It is well-known [15], [25], [30] that a certain portion of the necessary conditions may be obtained

*Adapted from a paper presented at the Fall Meeting of the Western Section, ORSA on 21 Oct. 1970 at Los Angeles, California.

†This research was supported by the Office of Naval Research as part of the Foundation Research Program at the Naval Postgraduate School.

by use of a squared variable technique, but that this result is not, in itself, particularly useful (see also [12], [14], [18] and [31]). Klein's approach [30] to create an equivalent equality constrained problem was motivated by Valentine's work [40] in the calculus of variations. In this latter field the approach has proven to be very useful (see [4], [5], [6] and references contained therein). We show here that additional results can be readily obtained in nonlinear programming when the well-known second order conditions for equality constrained problems (see pp. 115–116 of [27], pp. 191–195 of [13], [10], pp. 122–136 of [22], pp. 37–53 of [25], pp. 52–56 of [33]) are considered. However, as is well-known, we must make certain assumptions which qualify the problem under consideration, and we discuss the constraint qualification that we employ in the next section.

2. THE NORMALITY ASSUMPTION

Our approach is to convert an optimization problem with inequalities and nonnegative variables to an equivalent equality-contrained problem with unrestricted variables by means of the well-known "squared-variable transformation." We then apply the classical Lagrange multiplier theorem to determine the stationary points of this equivalent problem and further examine these with second order conditions. The usual assumption made in order that the Lagrange multiplier method yield necessary conditions for an extremum is that the Jacobian matrix of the constraints has full row rank at a locally optimal point [2]. Gould and Tolle have recently shown that a much weaker assumption is possible in this classical case [24]; theirs is, in fact, the weakest assumption possible.

Let us denote the set of points satisfying the constraints of the optimization problem by S. We shall refer to a point $x \epsilon S$ as a normal point of S if the constraint gradients are linearly independent at x. Following Bliss (see pp. 210–213 of [8]), we shall refer to a point which is not normal as "abnormal." We henceforth assume in this paper that all local optima occur at normal points. It is well known that this normality assumption is stronger than the constraint qualification used by Kuhn and Tucker [3], [24]. It is also well known that the Lagrange multipliers are not unique in the abnormal case (see 67–68 of [26]).

The reader should note, however, that we make the normality assumption for the equivalent equality-constrained problem with squared variables. For the original inequality-constrained problem, this is equivalent to assuming that the gradients of those constraints which are binding at the point in question are linearly independent and that the problem is "nondegenerate" (in the sense that its solution contains at least as many nonzero variables as there are binding constraints at the point in question). For example, consider a linear programming problem. It may be shown that a degenerate optimal solution to a linear program is necessary and sufficient for an optimal point of the corresponding squared-variable problem to be "abnormal" when all the constraints of the original problem are linearly independent. Hence, our present development is inadequate in this instance. A similar remark applies in the nonlinear case.

3. THE SPECIAL CASE OF LINEAR PROGRAMMING

We originally discovered our results in attempting a calculus approach to linear programming, although these methods are, of course, more general. It seems appropriate to consider this special case separately, since it motivates our later developments and certain arguments to be used later are simpler. Also, the development of linear programming results via the squared-variable transformation approach does not appear to be adequately discussed in the literature. One of the more thorough developments is by Charnes and Cooper (see pp. 662–667 of [15]). However, because they do

not consider second order conditions, they conclude that (p. 666 of [15]) "all this classical analysis has achieved is to lead us back to the beginning of linear-programming methods." G. Dantzig (see pp. 140–144 of [19]) incorrectly developed some linear programming results by classical Lagrange multiplier methods (after applying a squared-variable transformation), since he did not recognize that the bordered Hessian condition (or its equivalent) must be employed for sufficiency if one maintains the Lagrangian approach.

After an initial version of this present paper had been written, the paper of Taha and Curry [38] appeared. These authors used an approach that is slightly different from the one employed here and, moreover, their basic results contain certain gaps as noted elsewhere [37]. In particular, they made the normality assumption for an equivalent squared-variable problem, but failed to note that this assumption is not satisfied when the original linear program has a degenerate optimal solution. We have attempted to discuss such points more thoroughly in the present development.

We consider*

$$\text{maximize } \sum_{i=1}^{n} c_i x_i,$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad \text{for } i = 1, \ldots, m$$

$$x \geq 0. \tag{LP}$$

An equivalent problem is

$$\text{maximize } \sum_{i=1}^{n} c_i u_i^2,$$

subject to

$$g_i(u, s) = \sum_{j=1}^{n} a_{ij} u_j^2 + s_i^2 - b_i = 0 \qquad \text{for } i = 1, \ldots, m. \tag{LPE}$$

The Lagrangian, $\mathscr{L}(u, s, \lambda)$, is given by

$$\mathscr{L}(u, s, \lambda) = \sum_{i=1}^{n} c_i u_i^2 - \sum_{j=1}^{m} \lambda_j \left\{ \sum_{k=1}^{n} a_{jk} u_k^2 + s_j^2 - b_j \right\}.$$

We assume that the maximum to (LPE) does not occur at an "abnormal" point. In this case, necessary conditions for a maximum are given by

*The notation $x \geq 0$ denotes that each component of the vector x is greater than or equal to zero, i.e., $x_i \geq 0$ for $1, \ldots, n$.

$$\frac{\partial \mathscr{L}}{\partial u_i}=0=2u_i\left\{c_i-\sum_{j=1}^{m}\lambda_j a_{ji}\right\} \qquad \text{for } i=1,\ \ldots,\ n, \tag{1}$$

$$\frac{\partial \mathscr{L}}{\partial s_i}=0=-2\lambda_i s_i \qquad \text{for } i=1,\ \ldots,\ m, \tag{2}$$

$$\frac{\partial \mathscr{L}}{\partial \lambda_i}=0=-\left\{\sum_{k=1}^{n} a_{ik}u_k^2+s_i^2-b_i\right\} \qquad \text{for } i=1,\ \ldots,\ m. \tag{3}$$

A point $P^*=(u^*, s^*, \lambda^*)$, having $2m+n$ components, at which the above $2m+n$ necessary conditions are satisfied is called a stationary point. It may be shown (see p. 666 of [15]) than an extreme point of the feasible region is a stationary point.

As noted above (also see pp. 152–156 of [2] or pp. 96–100 of [27]) the conditions (1)–(3) are only necessary at a normal point of (LPE) where at least one Jacobian of the m constraints with respect to m of the $m+n$ variables does not vanish. Let us consider the mth order Jacobian with respect to the variables $u_1, \ldots, u_{m-r}$ and $s_{m-r+1}, \ldots, s_m$. Then the Jacobian of interest for the equivalent problem (LPE), $J=\dfrac{\partial(g_1, \ldots, g_m)}{\partial(u_1, \ldots, s_m)}$, may be shown to be given by

$$J=2^m\left(\prod_{i=1}^{m-r} u_i\right)\left(\prod_{i=m-r+1}^{m} s_i\right)\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1,\,m-r} \\ a_{21} & a_{22} & \cdots & a_{2,\,m-r} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{m-r,\,1} & a_{m-r,\,2} & \cdots & a_{m-r,\,m-r} \end{vmatrix}, \tag{4}$$

where the first $(m-r)$ constraints of (LP) have been assumed to be linearly independent. The index r denotes the number of nonbinding constraints, i.e., the corresponding slack variables are all nonzero, so that when all constraints are binding, then $r=0$. Considering (4), we see that in order for P^* to be a normal point of (LPE) at least m of the u^*'s and s^*'s must be nonzero. Hence, a linear program (LP) with a degenerate optimal solution implies that the equivalent problem (LPE) has a maximum at an "abnormal" point. It may also be shown that the converse is true.

A sufficient condition for a maximum to the above problem is that all roots of a certain polynomial be negative (see pp. 115–116 of [27]). It is convenient to relabel the variables $u_1, \ldots, u_n, s_1, \ldots, s_m$ as $y_1, \ldots, y_n, y_{n+1}, \ldots, y_{n+m}$. In this notation we have

$$L_{ij}=\frac{\partial^2 \mathscr{L}}{\partial y_i \partial y_j}=\begin{cases} 2\left\{c_i-\sum_{k=1}^{m}\lambda_k a_{ki}\right\} & \text{for } i=j=1,\ \ldots,\ n, \\ -2\lambda_i & \text{for } i=j=n+1,\ \ldots,\ n+m, \\ 0 & \text{for } i\neq j, \end{cases}$$

and

$$g_{ij}=\frac{\partial g_i}{\partial y_j}=\begin{cases}2u_j a_{ij} & \text{for } i=1,\ldots,m \text{ and } j=1,\ldots,n,\\ 2s_{j-n}\delta_{i,j-n} & \text{for } i=1,\ldots,m \text{ and } j=n+1,\ldots,n+m,\end{cases}$$

where δ_{ij} is equal to one for $i=j$ and zero otherwise. For a sufficient condition we consider the following determinantal equation

$$\begin{vmatrix}
L_{11}-e & 0 & \cdots & 0 & 0 & \cdots & 0 & 2u_1a_{11} & \cdots & 2u_1a_{m1}\\
0 & L_{22}-e & \cdots & 0 & 0 & \cdots & 0 & 2u_2a_{12} & \cdots & 2u_2a_{m2}\\
\cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & & \cdot\\
0 & 0 & \cdots & L_{nn}-e & 0 & \cdots & 0 & 2u_na_{1n} & \cdots & 2u_na_{mn}\\
0 & 0 & \cdots & 0 & -2\lambda_1-e & \cdots & 0 & 2s_1 & \cdots & 0\\
\cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot\\
\cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot\\
\cdot & \cdot & & \cdot & \cdot & \cdot & 0 & \cdot & & \cdot\\
0 & 0 & \cdots & 0 & 0 & \cdots & -2\lambda_m-e & 0 & \cdots & 2s_m\\
2u_1a_{11} & 2u_2a_{12} & \cdots & 2u_na_{1n} & 2s_1 & \cdots & 0 & 0 & \cdots & 0\\
\cdot & \cdot & & \cdot & 0 & & \cdot & \cdot & & \cdot\\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & & \cdot\\
 & & & & \cdot & & & & & \\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & & \cdot\\
2u_1a_{m1} & 2u_2a_{m2} & \cdots & 2u_na_{mn} & 0 & \cdots & 2s_m & 0 & \cdots & 0
\end{vmatrix}=0. \tag{5}$$

Let us denote Equation (5) as $P(e)=0$, where $P(e)$ is an nth degree polynomial in e. The sufficient condition that an isolated local maximum occur at a stationary point $P^*=(u^*, s^*, \lambda^*)$ is that at this point all n roots of $P(e)=0$ are negative.

We now show how the above classical theory readily yields the well known necessary and sufficient conditions for a maximum to (LP) in the nondegenerate case. We consider necessity first. Considering the derivation of the bordered Hessian condition, it is easily seen (see also Theorem 4 on pp. 645–646 of [34]) that in order for a local maximum to occur at a normal point, it is necessary that P^* be a stationary point and that all roots of $P(e)=0$ be nonpositive.

From the necessary condition (2), we have that when $\lambda_i \neq 0$, then $s_i=0$ so that in (5) all entries in the $(n+i)$th column are zero except for the term $(-2\lambda_i-e)$. Hence

$$P(e)=(-2\lambda_i-e)P_1(e)=0,$$

and $\lambda_i \geq 0$. Considering the necessary condition (1), we have that if $c_i - \sum_{j=1}^{m} \lambda_j a_{ji} \neq 0$, then $u_i = 0$ so that a similar argument yields that

$$\sum_{j=1}^{m} \lambda_i a_{ij} \geq c_i.$$

Thus, we have obtained the well-known necessary conditions for a nondegenerate maximal solution to (LP)

$$\begin{aligned} Ax^* \leq b \qquad & A^T\lambda^* \geq c \\ x^* \geq 0 \qquad & \lambda^* \geq 0 \\ x_i^* t_i = 0 \qquad & \lambda_i^* s_i = 0, \end{aligned} \tag{6}$$

where s_i and t_i are slack variables to the primal and dual problems, respectively, and x is a column vector.

The sufficient condition that a normal point, which is also a stationary point satisfying (1)–(3), be a local isolated maximum to (LPE) is that all n roots of $P(e) = 0$ be negative. For a point to be normal, satisfy (1)–(3) and this sufficiency condition, it is easily shown that it is sufficient that there are exactly m nonzero u^*'s and s^*'s, precisely one term of any complementary slackness relationship (1) or (2) is zero, and (6) holds. We show below that for the isolated maximum sufficiency condition to be satisfied at a normal point, it is necessary that there are precisely m variables nonzero. Consider the case when $u_1, \ldots, u_{m-r}$ and $s_{m-r+1}, \ldots, s_m$ as determined by (1)–(3) are nonzero. Then, it may be shown that (5) is given by

$$\prod_{i=m-r+1}^{n} (L_{ii} - e) \cdot \prod_{i=1}^{m-r} (-2\lambda_i - e) \cdot 2^{2m} \left(\prod_{i=1}^{m-r} u_i \right)^2 \left(\prod_{i=m-r+1}^{m} s_i \right)^2 \cdot \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1,m-r} \\ a_{21} & a_{22} & \cdots & a_{2,m-r} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{m-r,1} & a_{m-n,2} & \cdots & a_{2,m-r} \end{vmatrix}^2 = 0, \tag{7}$$

where the first $(m-r)$ constraints of (LP) have been assumed to be linearly independent, and we have made use of some properties of bordered determinants (see pp. 26–30 of [9] or pp. 17–18 and pp. 29–30 of [11]). Hence, the existence of exactly m non-zero u^*'s and s^*'s satisfying (1)–(3) and (6) such that $\lambda_1^* > 0$ if $s_i^* = 0$ and $\sum_{j=1}^{m} \lambda_j^* a_{ij} > c_i$ if $x_i^* = u_i^{*2} = 0$ is sufficient for an isolated local maximum, which is also seen to be global, since we have a concave objective function subject to convex constraints (see p. 93 of [26]).

We note that the above classical sufficient condition for an isolated maximum is not satisfied when $\sum_{j=1}^{m} \lambda_j^* a_{ji} = c_i$ corresponding to $u_i^* = 0$, since then $P(e) = 0$ has a zero root. This happens when there

are alternate optima to (LP), and we examine the second order conditions for (LPE) at an extreme point of the feasible region where the maximum occurs. However, as Taha and Curry have pointed out [38], in the special case of a quadratic objective function and quadratic constraints (such as (LPE)), it is sufficient for a (not necessarily isolated) maximum to occur at a stationary point of (LPE) that all n roots of $P(e)=0$ be nonpositive (i.e., either zero or negative).

We now consider further the case when there are alternate optima to (LP). Let us focus on the case when a non-extreme point satisfies (1)–(3). Then we have a stationary point P^* with more than m nonzero u^*'s and s^*'s (for simplicity assume that all constraints are binding). Let $u_1^*, \ldots, u_{m+r}^*$ be nonzero. It may be shown that (5) becomes

$$\prod_{i=m+r+1}^{n} (L_{ii}^*-e)\cdot \prod_{i=1}^{m} (-2\lambda_i^*-e)\cdot (-1)^r e^r \left(\frac{1}{m}\right) 2^{2m}.$$

(8)
$$\left[\sum_{i_1,\ldots,i_m=1}^{m+r} \left\{ \begin{vmatrix} a_{1i_1} & a_{1i_2} & \cdots & a_{1i_m} \\ a_{2i_1} & a_{2i_2} & \cdots & a_{2i_m} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{mi_1} & a_{mi_2} & \cdots & a_{mi_m} \end{vmatrix} \left(\prod_{k=1}^{m} u_{i_k}^* \right) \right\}^2 \right] = 0,$$

where additional properties of bordered determinants have been used (see p. 37 of [11]). Hence, it is sufficient for a nonisolated maximum to occur at a stationary point of (LPE) that $\sum_{j=1}^{m} \lambda_j^* a_{ji} \geq c_i$ for $i=m+r+1, \ldots, n$ and $\lambda_i^* \geq 0$ for $i=1, \ldots, m$. We observe that by (8) $e=0$ is a root of multiplicity r of $P(e)=0$. Thus, we have obtained the well-known result that conditions (6) are necessary and sufficient for a nondegenerate maximal solution to (LP).

Moreover, we now show that in order for the second order sufficiency condition for an isolated maximum to be satisfied at a normal point, then it is necessary that there are precisely m nonzero u^*'s and s^*'s satisfying (1)–(3). Above, we have just shown that more than m nonzero variables at a stationary point leads to $e=0$ being one of the n roots of $P(e)=0$ so that the sufficient condition for an isolated maximum is not satisfied. If there are less than m nonzero variables at a stationary point of (LPE), then the point is not normal by (4).

It also seems appropriate to note that the classical Lagrange multiplier approach applied to an equivalent equality constrained problem yields more than conditions (6), since these have implicitly in them the well-known linear programming result that a sufficient condition for a basic feasible solution to be a maximal feasible solution is that $z_j - c_j \geq 0$ (see p. 61 of [23]). To avoid notational difficulties we recast the basic problem in a slightly different form: we let all the constraints of (LP) be equalities, thus incorporating the s's into the u's to have $m+n$ u's in (LPE).

By similar application of our classical approach, it readily follows that necessary conditions for a maximum at a normal point $x_i^* = u_i^{*2}$ of (LPE) is that

$$Ax^* = b \qquad A^T\lambda^* \geq c$$

$$x^* \geq 0$$

$$x_i^* \left\{ \sum_{j=1}^{m} a_{ji}\lambda_j^* - c_i \right\} = 0 \qquad \text{for } i = 1, \ldots, m+n. \tag{9}$$

Following previous arguments, it is easily shown that the above conditions are sufficient for a maximal solution, the maximum being isolated when the inequality $\sum_{j=1}^{m} a_{ji}\lambda_j^* - c_i > 0$ holds for exactly n indices i. Let us denote by x_B^* those m x^*'s not equal to zero. Then we may delete those columns of A for which the corresponding $x_i^* = 0$ to obtain

$$Bx_B^* = b,$$

where B is an $m \times m$ nonsingular matrix. Multiplying expressions in (9) by appropriate quantities and summing, we obtain

$$(\lambda^*, b) = (c, x^*) = (c_B, x_B^*) = (c_B, B^{-1}b) = ((B^{-1})^T c_B, b),$$

where $(a, b) = \sum_{i=1}^{n} a_i b_i$ is an inner product, and we have made use of the bilinear identity $(y, Ax) = (A^T y, x)$. Thus, the classical Lagrange multiplier method has yielded that for a nondegenerate maximum it is sufficient that for x^* satisfying (9) we have

$$A^T(B^{-1})^T c_B \geq c,$$

which is the familiar linear programming result. By considering elementary properties of linear inequalities, this well-known result is easily established for more general conditions (i.e., for degenerate solutions).

4. A NONLINEAR PROGRAMMING PROBLEM

We consider

$$\text{maximize } f(x_1, \ldots, x_n),$$

subject to

$$g_i(x_1, \ldots, x_n) \leq 0 \qquad \text{for } i = 1, \ldots, m$$

$$x \geq 0. \tag{NLP}$$

We assume that all functions are twice continuously differentiable in all arguments. An equivalent problem is

$$\text{maximize } f(u_1^2, \ldots, u_n^2),$$

subject to

$$g_i(u_1^2, \ldots, u_n^2)+s_i^2=0 \qquad \text{for } i=1, \ldots, m. \tag{NLPE}$$

The Lagrangian, $\mathscr{L}(u, s, \lambda)$, is given by

$$\mathscr{L}(u, s, \lambda)=f(u_1^2, \ldots, u_n^2)-\sum_{j=1}^{m} \lambda_j\{g_j(u_1^2, \ldots, u_n^2)+s_j^2\}.$$

We assume that all the relative maxima of (NLPE) occur at normal points. In this case, necessary conditions for a local maximum are given by

$$\frac{\partial \mathscr{L}}{\partial u_i}=0=2u_i\left\{\frac{\partial f}{\partial x_i}-\sum_{j=1}^{m} \lambda_j \frac{\partial g_j}{\partial x_i}\right\} \qquad \text{for } i=1, \ldots, n, \tag{10}$$

$$\frac{\partial \mathscr{L}}{\partial s_i}=0=-2\lambda_i s_i \qquad \text{for } i=1, \ldots, m, \tag{11}$$

$$\frac{\partial \mathscr{L}}{\partial \lambda_i}=0=-g_i(u_1^2, \ldots, u_n^2)-s_i^2 \qquad \text{for } i=1, \ldots, m. \tag{12}$$

As before, we refer to a point $P^*=(u^*, s^*, \lambda^*)$, having $2m+n$ components, at which the above $2m+n$ necessary conditions are satisfied as a stationary point. At a stationary point P^* a typical mth order Jacobian, one of which must be nonvanishing by the normality assumption, is given by

$$J=2^m\left(\prod_{i=1}^{m-r} u_i^*\right)\left(\prod_{i=m-r+1}^{m} s_i^*\right)\begin{vmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_{m-r}} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_{m-r}} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{\partial g_{m-r}}{\partial x_1} & \frac{\partial g_{m-r}}{\partial x_2} & \cdots & \frac{\partial g_{m-r}}{\partial x_{m-r}} \end{vmatrix}$$

In considering second order conditions, it is convenient to relabel the variables $u_1, \ldots, u_n$, $s_1, \ldots, s_m$ as $y_1, \ldots, y_n, y_{n+1}, \ldots, y_{n+m}$. In this notation we have

$$L_{ij}=\frac{\partial^2 \mathscr{L}}{\partial y_i \partial y_j}=\begin{cases} 4u_iu_j\left\{\dfrac{\partial^2 f}{\partial x_i \partial x_j}-\displaystyle\sum_{k=1}^{m} \lambda_k \frac{\partial^2 g_k}{\partial x_i \partial x_j}\right\}+2\delta_{ij}\left\{\dfrac{\partial f}{\partial x_i}-\displaystyle\sum_{k=1}^{m} \lambda_k \frac{\partial g_k}{\partial x_i}\right\} & \text{for } i, j=1, \ldots, n, \\ -2\delta_{ij}\lambda_{i-n} & \text{for } i, j=n+1, \ldots, n+m, \\ 0 & \text{otherwise} \end{cases}$$

and

$$g_{ij}=\frac{\partial g_i}{\partial y_j}=\begin{cases} 2u_j\dfrac{\partial g_i}{\partial x_j} & \text{for } i=1,\ldots,m \text{ and } j=1,\ldots,n, \\ 2s_{j-n}\delta_{i,j-n} & \text{for } i=1,\ldots,m \text{ and } j=n+1,\ldots,n+m. \end{cases}$$

The second order conditions lead us to consider the following determinantal equation

$$\begin{vmatrix} L_{11}-e & L_{12} & \ldots & L_{1n} & 0 & \ldots & 0 & g_{11} & g_{21} & \ldots & g_{m1} \\ L_{21} & L_{22}-e & \ldots & L_{2n} & 0 & \ldots & 0 & g_{12} & g_{22} & \ldots & g_{m2} \\ \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ L_{n1} & L_{n2} & \ldots & L_{nn}-e & 0 & \ldots & 0 & g_{1n} & g_{2n} & \ldots & g_{mn} \\ 0 & 0 & \ldots & 0 & -2\lambda_1-e & \ldots & 0 & 2s_1 & 0 & \ldots & 0 \\ \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & \ldots & 0 & 0 & \ldots & -2\lambda_m-e & 0 & 0 & \ldots & 2s_m \\ g_{11} & g_{12} & \ldots & g_{1n} & 2s_1 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ g_{21} & g_{22} & \ldots & g_{2n} & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ g_{m1} & g_{m2} & \ldots & g_{mn} & 0 & \ldots & 2s_m & 0 & 0 & \ldots & 0 \end{vmatrix} = 0. \tag{13}$$

Let us denote Equation (13) as $P(e)=0$, where $P(e)$ is an nth degree polynomial in e. As before a necessary condition that a local maximum occur at a stationary point P^*, which is a normal point, is that all the n roots of $P(e)=0$ be nonpositive. A sufficient condition for an isolated local maximum to occur at P^* is that all the n roots of $P(e)=0$ be negative.

At a stationary point let us arrange the y's so that the r y's corresponding to the $r \leq n$ x's which are positive appear first and the s y's corresponding to the $s \leq m$ nonzero slack variables appear after those corresponding to the zero slack variables. From the complementary slackness relationships (10) and (11) and an argument similar to that used in the linear programming case, we obtain

$$P(e)=0=\prod_{i=r+1}^{n}(L_{ii}-e)\cdot\prod_{i=1}^{m-s}(-2\lambda_i-e)\cdot D(e), \tag{14}$$

where it may be shown that

$$D(e) = \begin{vmatrix}
L'_{11}-e'_1 & L'_{12} & \cdots & L'_{1r} & 0 & \cdots & 0 & g'_{11} & g'_{21} & \cdots & g'_{m1} \\
L'_{21} & L'_{22}-e'_2 & \cdots & L'_{2r} & 0 & \cdots & 0 & g'_{12} & g'_{22} & \cdots & g'_{m2} \\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\
L'_{r1} & L'_{r2} & \cdots & L'_{rr}-e'_r & 0 & \cdots & 0 & g'_{1r} & g'_{2r} & \cdots & g'_{mr} \\
0 & 0 & \cdots & 0 & -e & \cdots & 0 & 2s_{m-s+1} & 0 & \cdots & 0 \\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & & & \cdot \\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\
0 & 0 & \cdots & 0 & 0 & \cdots & -e & 0 & 0 & \cdots & 2s_m \\
g'_{11} & g'_{12} & \cdots & g'_{1r} & 2s_{m-s+1} & \cdots & 0 & 0 & 0 & \cdots & 0 \\
g'_{21} & g'_{22} & \cdots & g'_{2r} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot \\
g'_{m1} & g'_{m2} & \cdots & g'_{mr} & 0 & \cdots & 2s_m & 0 & 0 & \cdots & 0
\end{vmatrix}$$

and further

$$-e'_i = \frac{e}{4x_i} \qquad \text{for } i = 1, \ldots, r,$$

$$L'_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} - \sum_{k=1}^{s} \lambda_k \frac{\partial^2 g_k}{\partial x_i \partial x_j} \qquad \text{for } i, j = 1, \ldots, r,$$

and

$$g'_{ij} = \frac{\partial g_i}{\partial x_j} \qquad \text{for } i = 1, \ldots, m \text{ and } j = 1, \ldots, r.$$

It may be shown using properties of bordered determinants that $D4e)$ is a polynomial of degree $r+s-m$ in e, where $r+s \geq m$ (so that some mth order Jacobian is nonvanishing).

The second order necessary condition as applied to (14) and (10)–(12) readily yields the well-known necessary (Kuhn-Tucker) conditions for a local maximum of (NLP) (assuming that the corresponding point of (NLPE) is normal).

$$\frac{\partial f}{\partial x_i} - \sum_{k=1}^{m} \lambda_k^* \frac{\partial g_k}{\partial x_i} \leq 0$$

(15)
$$x_i^* \left\{ \frac{\partial f}{\partial x_i} - \sum_{k=1}^{m} \lambda_k^* \frac{\partial g_k}{\partial x_i} \right\} = 0$$

$$x_i^* \geq 0 \qquad \text{for } i = 1, \ldots, n,$$

$$g_k \leq 0$$

$$\lambda_k^* g_k = 0$$

$$\lambda_k^* \geq 0 \qquad \text{for } k = 1, \ldots, m.$$

When there are precisely m nonzero variables (both x's and slack variables), then (15) plus consideration of (14) give sufficient conditions for an isolated local maximum (i.e. (15) holds with $\lambda_i^* > 0$ when $s_i^* = 0$ for $i = 1, \ldots, m$ and

$$\frac{\partial f^*}{\partial x_i} - \sum_{j=1}^{m} \lambda_j^* \frac{\partial g_j^*}{\partial x_i} < 0$$

when $x_i^* = 0$). When $r + s > m$, then (15) plus that all the $(r+s-m)$ roots of $D(e) = 0$ must be negative and consideration of (14) give sufficient conditions for an isolated local maximum. We also note that when the objective function is concave and the constraint functions are convex (so that the constraint set $S = \{x | g_i(x) \leq 0 \text{ for } i = 1, \ldots, m\}$ is convex), the above are sufficient conditions for a global maximum (see p. 93 of [26]).

It should also be noted that when $f(x)$ is concave and all the $g_i(x)$ are convex, then a sufficient condition for an isolated relative (global) maximum of $f(x)$ at P^* is that (15) holds with $\lambda_i^* > 0$ when

$$s_i^* = 0 \qquad \text{for } i = 1, \ldots, m \qquad \text{and } \frac{\partial f^*}{\partial x_i} - \sum_{j=1}^{m} \lambda_j^* \frac{\partial g_j^*}{\partial x_i} < 0 \qquad \text{when } x_i^* = 0.$$

We assume that either the objective function is strictly concave or at least one of the binding constraint functions is strictly convex. The proof is as follows. Let s be the number of nonbinding constraints. No more than $(n-m+s)$ of the x^*'s can be zero for P^* to be a normal point. If exactly $(n-m+s)$ x^*'s are zero, then by a previous argument the second order sufficiency condition is satisfied. If less than $(n-m+s)$ of the x^*'s are zero, then it may be shown that all the $(r+s-m)$ roots of $D(e)$ are negative (after deletion from $P(e)$ of terms corresponding to the $(m-s)$ zero s^*'s and the zero u^*'s). This follows from the convexity assumptions, nonnegativity of the multipliers, and consideration of the derivation of the bordered Hessian condition from maximization of a quadratic form with general term $L_{ij} dy_i dy_j$ for all dy's satisfying differential constraints. If the unconstrained quadratic form is negative definite, then adding side conditions will not change this.

5. DISCUSSION

We believe that the present derivation of the Kuhn-Tucker conditions is important because it shows that the classical theory of equality constrained optimization may be readily extended to linear and nonlinear programming. The fact that this extension is possible may generate new interest in classical methods. Second order conditions play a central role in our development. Economists are quite familiar (see [1], [10] and [33] and references contained therein) with the classical second order conditions (bordered Hessian).

The approach shown in this paper may be applied to problems with both nonnegative and un-

restricted variables and both equality and inequality constraints. For example, the results obtained by Dorn [20] (Theorems 1, 2 and 3) are readily deducible from consideration of the bordered Hessian conditions.

We did not assume a weaker constraint qualification for reasons of simplicity. Gould and Tolle [24] have established the weakest possible "first order constraint qualification" (see pp. 643–645 of [34]) for the classical equality-constrained problem. Recall that we make the constraint qualification assumption for the equivalent equality-constrained problem. If we assumed a weaker constraint qualification for this problem, then we would have to analyze what assumption this implies for the original inequality-constrained problem. We should also note that a second order constraint qualification is required by the second order necessary condition [34]. Our normality assumption implies that this is met.

ACKNOWLEDGEMENT

The author wishes to express his appreciation to W. Raike, who critically read the manuscript and offered corrections for several technical errors in earlier versions. He also made numerous helpful comments and suggestions. The referee kindly pointed out the reference to Afriat's recent work [1].

REFERENCES

[1] Afriat, S., "Theory of Maxima and the Method of Lagrange," SIAM J. of Appl. Math. *20*, 343–357 (1971).

[2] Apostol, T., *Mathematical Analysis* (Addison-Wesley Publishing Co., Inc., Reading, Massachusetts, 1957).

[3] Arrow, K., L. Hurwicz, and H. Uzawa, "Constraint Qualifications in Maximization Problems," Nav. Res. Log. Quart. *8*, 175–191 (1961).

[4] Berkovitz, L., "Variational Methods in Problems of Control and Programming," J. Math. Analysis and Applications *3*, 145–169 (1961).

[5] Berkovitz, L., "On Control Problems with Bounded State Variables," J. Math. Analysis and Applications *5*, 488–498 (1962).

[6] Berkovitz, L., "A Variational Approach to Differential Games," in *Advances in Game Theory* (M. Dresher, L. Shapley, A. Tucker, eds.) (Princeton University Press, Princeton, New Jersey, 1964), pp. 127–174.

[7] Bernholtz, B., "A New Derivation of the Kuhn-Tucker Conditions," Operations Research *12*, 295–299 (1964).

[8] Bliss, G., *Lectures on the Calculus of Variations* (University of Chicago Press, Chicago, 1946).

[9] Bôcher, M., *Introduction to Higher Algebra* (The Macmillan Company, New York, 1907) (available as Dover reprint).

[10] Burger, E., "On Extrema with Side Conditions," Econometrica *23*, 451–452 (1955).

[11] Burnside, W. and A. Panton, *The Theory of Equations* (Longmans, Green, and Company, London, 1928) Vol. 2 (available as a Dover reprint).

[12] Camp, G., "Inequality-Constrained Stationary-Value Problems," Operations Research *3*, 548–550 (1955).

[13] Carathéodory, C., *Variationsrechnung und Partielle Differential gleichungen erster Ordung*, Teubner, Leipzig (1935), English translation *Calculus of Variations and Partial Differential Equations of the First Order* (Holden-Day, Inc., San Francisco, 1965, 1967).

[14] Charnes, A. and W. Cooper, "Such Solutions are Very Little Solved," Operations Research *3*, 345–346 (1955).

[15] Charnes, A. and W. Cooper, *Management Models and Industrial Applications of Linear Programming* (John Wiley & Sons, Inc., New York, 1961).

[16] Courant, R. and D. Hilbert, *Methods of Mathematical Physics* (Interscience Publishers, Inc., New York, 1953).

[17] Dantzig, G., "Maximization of a Linear Function of Variables Subject to Linear Inequalities," pp. 339–347 in Activity Analysis of Production and Allocation (T. Koopmans, ed.) (John Wiley & Sons, Inc., New York, 1951).

[18] Dantzig, G., "Note on Klein's 'Direct Use of External Principles in Solving Certain Problems Involving Inequalities,'" Operations Research *4*, 247–249 (1956).

[19] Dantzig, G., *Linear Programming and Extensions* (Princeton University Press, Princeton, New Jersey, 1963).

[20] Dorn, W., "On Lagrange Multipliers and Inequalities," Operations Research *9*, 95–104 (1961).

[21] Farkas, J., "Über die Theorie der einfachen Ungleichungen," Journal für die reine und angewandte Mathematik *124*, 1–27 (1901).

[22] Frisch, R., *Maxima and Minima* (Dordrecht-Holland, 1966).

[23] Gass, S., *Linear Programming*, (McGraw-Hill, New York, 1964) Second Edition.

[24] Gould, F. and J. Tolle, "A Necessary and Sufficient Qualification for Constrained Optimization," SIAM J. Appl. Math. *20*, 164–172 (1971).

[25] Gue, R. and M. Thomas, *Mathematical Methods in Operations Research* (The Macmillan Company, New York, 1968).

[26] Hadley, G., *Nonlinear and Dynamic Programming* (Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1964).

[27] Hancock, H., *Theory of Maxima and Minima* (Ginn and Company, New York, 1917) (available as Dover reprint).

[28] John, F., "Extremum Problems with Inequalities as Subsidiary Conditions," Courant Anniversary Volume (Interscience, New York, 1948), pp. 187–204.

[29] King, R., "Necessary and Sufficient Conditions for Inequality Constrained Extreme Values," Industrial and Engineering Chemistry Fundamentals *5*, 484–489 (1966).

[30] Klein, B., "Direct Use of Extremal Principles in Solving Certain Optimizing Problems Involving Inequalities," Operations Research *3*, 168–175 (1955).

[31] Klein, B., "Further Remarks on the Direct Use of Extremal Principles in Solving Certain Optimizing Problems Involving Inequalities," Operations Research *3*, 548 (1955).

[32] Kuhn, H. and A. Tucker, "Nonlinear Programming," Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability (J. Neyman, ed.) (University of California Press, Berkeley, California, 1951), pp. 481–492.

[33] Lancaster, K., *Mathematical Economics* (The Macmillan Company, New York, 1968).

[34] McCormick, G., "Second Order Conditions for Constrained Minima," SIAM J. Appl. Math. *15*, 641–652 (1967).

[35] Minkowski, H., *Geometrie der Zahlen* (B. G. Teubner, Leipzig, 1910).

[36] von Neumann, J., "Discussion of a Maximum Problem," John von Neumann, Collected Works, Volume VI (A. Taub, ed.) (Pergamon Press, Oxford, 1963), pp. 89–95.

[37] Raike, W. and J. Taylor, "Comments on the 'Classical' Derivation by Taha and Curry of Optimality Conditions in Linear Programming," to appear, Operations Research.

[38] Taha, H. and G. Curry, "Classical derivation of the Necessary and Sufficient Conditions for Optimal Linear Programs," Operations Research *19*, 1045–1050 (1971).

[39] Tucker, A., "Linear and Nonlinear Programming," Operations Research *5*, 244–257 (1957).

[40] Valentine, F., "The Problem of Lagrange with Differential Inequalities as Added Side Conditions," Contributions to the Calculus of Variations, 1933–1937 (The University of Chicago Press, Chicago, Illinois, 1937), pp. 403–447.

[41] Wilde, D., "Differential Calculus in Nonlinear Programming," Operations Research *10*, 764–773 (1962).

[42] Wilde, D., "Jacobians in Constrained Nonlinear Optimization," Operations Research *13*, 848–856 (1965).

PARTIAL INFORMATION IN TWO PERSON GAMES WITH RANDOM PAYOFFS*

R. G. Cassidy

Carnegie-Mellon University
Pittsburgh, Pennsylvania

C. A. Field and M. J. L. Kirby

Dalhousie University
Halifax, Nova Scotia, Canada

I. INTRODUCTION

In a previous paper [4], we have considered the solution of one person random payoff games against nature in which the player had only partial information about the distribution function, $F(A)$, of the random payoff matrix A. In this paper we extend these results to two person games. At the same time we introduce new types of partial information which may be available to the players of the game.

By a two person random payoff game, we mean a two person zero sum game with $m \times n$ payoff matrix $A = \{a_{ij}\}$, where A is a random matrix with distribution function $F(A)$. In the game, player I has m ($i = 1, 2, \ldots m$) pure strategy alternatives, and player II has n ($j = 1, 2, \ldots n$) pure strategy alternatives. On any play of the game in which player I plays row i and player II plays column j, the actual payoff from player II to player I will be $a_{ij}(w)$, where w is selected from the domain of a_{ij} according to the marginal probability distribution of a_{ij}.

Before proceeding with a development of our models we briefly outline existing results on two persons games with incomplete information. In [7], [8], and [9] Harsanyi gives a definitive study of games with incomplete information. In his work, Harsanyi develops a whole structure of levels of partial information and presents a method of solution based on transforming the partial information to obtain the Bayesian equivalent complete information game. In effect, Harsanyi approaches the problem by considering it as one in which many different games may be played and the players must determine not only how to play a game, but also which game they should play. He solves the problem by assuming the players have an à priori distribution over the set of possible distribution functions $F(A)$ of the random payoff matrix A.

Banös in [1] considers partial information in games, but with a different approach from that of Harsanyi, since he assumes player II has complete information while player I has only partial information. Also Banös uses a sequential method of solution which enables player I to gain increased information about the payoff matrix as time goes on. Thus, asymptotically, player I can do as well as he would with complete information. In the development below, we consider only games where the information available to the players remains constant throughout the plays of the game.

In [4] we developed a model and a method of solution of a one person random payoff game against nature in which the player has only partial information about $F(A)$. In this development an expected value type of optimality criterion was used and we shall initially use the same type of optimality cri-

* The research underlying this report was partly supported by ONR Contract N00014-67-A-0126-0009, NRC grant A4024 and DRB grant 9701-18.

terion for the solution of two person games. However, because of the increased complexity of two person games, the methods of solution presented in this paper differ considerably from the methods of solution developed in [4]. Also, because of the presence of a second player in the games considered in this paper, several new models for the solution of random payoff games with partial information are developed. In these models, optimality criteria other than the expected value criteria are used.

By using the results of [2], we also consider "satisficing" type of optimality criterion as well as the expected value criterion. For example, the problem in which the player wishes to maximize the probability of winning a specified amount β gives use to the following linear program (see [2]).

(1.1) $$\max \alpha$$

subject to:

$$\sum_{i=1}^{m} X_i P(a_{ij} \geq \beta) \geq \alpha \qquad \forall_j$$

$$\sum_{i=1}^{m} X_i = 1, \qquad X_i \geq 0.$$

II. SOLUTION METHODS FOR TWO PERSON GAMES WITH PARTIAL ORDERING INFORMATION

Let A be an $m \times n$ payoff matrix for a two person random payoff game. Let $F(A)$ be the distribution function of the random matrix A. The states of nature $A(k)$ $k=1, 2, \ldots, K$ of A are the K possible values of A which can occur with nonzero probability. Thus we assume that A can take on only a finite number of distinct values.

Let $p(k)$ $k=1, 2, \ldots, K$, be the probability that $A(k)$, the kth state of nature of A, occurs. If all $p(k)$ $k=1, \ldots, K$ are known, then $F(A)$ is known and we say we have a two person random payoff game with complete information (see [2] and [3]). Suppose, however, that all $p(k)$ are not known. In particular, suppose that the players know a partial ordering of the following form as the states of nature of A:

$$k \leq k' \iff p(k) \leq p(k').$$

That is, the partial order relation $k \leq k'$ holds if and only if $p(k) \leq p(k')$. This type of partial information, which was considered in [4] and [6], results from the fact that the players may know that one state, say k', is more likely to occur than another state, k, without knowing explicit numerical values for the probability of either state.

Let Π be the set of all probability measures $P=(p(1), \ldots, p(k))$ compatible with the partial ordering information. That is

$$\Pi = \{P | p(k) \geq p(k') \iff k \geq k'\}.$$

We proceed by assuming that player I will adopt a pessimistic viewpoint and use a maximin optimality criterion in which he assumes that nature and player II will combine to give him as small a payoff as possible. In this case, the problem player I solves to determine his optimal strategy X is:

$$\max_X \min_Y \min_P \sum_{i=1}^{m} \sum_{j=1}^{n} x_i \left(\sum_{k=1}^{K} a_{ij}(k) p(k) \right) y_j$$

subject to

$$P \epsilon \Pi$$

(2.1)
$$\sum_{i=1}^{m} x_i = 1 \qquad x_i \geq 0 \qquad \forall i$$

$$\sum_{j=1}^{n} y_j = 1 \qquad y_j \geq 0 \qquad \forall j.$$

Model (2.1) is the direct extension to two person games of the models for one person games developed in [4]. However, (2.1) is considerably more complex than the models of [4] with the increased complexity being caused by the mixed strategy vector Y for player II. Although (2.1) appears to be a quadratic programming problem, we show below that (2.1) can be converted to an equivalent linear programming problem.

We note that an alternate approach would be to consider the linear program associated with the expected value game subject to the constraint that the distribution P belongs to Π; i.e.,

$$\max \alpha$$

subject to:
$$\sum_{i=1}^{m} x_i \left(\sum_{k=1}^{K} a_{ij}(k) p(k) \right) \geq \alpha \qquad \forall j$$

(2.2)
$$P \epsilon \Pi$$

$$\sum_{i=1}^{m} x_i = 1$$

$$x_i \geq 0 \qquad \forall i.$$

(2.2) is a quadratic programming problem in the variables x_i, $i = 1, \ldots, m$ and $p(k)$, $k = 1, \ldots, K$ and so can be solved using any of the existing algorithms for such problems. We will not discuss these algorithms here. Another alternative is to assume that the probability measures $P \epsilon \Pi$ are under the control of player I and so the optimal probability measure P^* selected will be the most favorable P from the point of view of player I. This leads to the problem:

$$\max_{X} \max_{P} \min_{Y} \sum_{i=1}^{m} \sum_{j=1}^{n} x_i \left(\sum_{k=1}^{K} a_{ij}(k) p(k) \right) y_j$$

subject to

$$\sum_{i=1}^{m} x_i = 1 \qquad x_i \geq 0 \qquad \forall i$$

(2.3)
$$\sum_{j=1}^{n} y_j = 1 \qquad y_j \geq 0 \qquad \forall j$$

$$P \epsilon \Pi$$

In addition to the fact that (2.3) represents an overly optimistic way of approaching the problem, it has the additional disadvantage of not reducing to a linear program as does (2.1).

We begin by observing that for fixed $P\epsilon\Pi$, (2.1) is a deterministic two person zero sum game with $m \times n$ payoff $B(P) = \{b_{ij}(P)\}$, where

$$b_{ij}(P) = \sum_{k=1}^{K} a_{ij}(k)p(k).$$

Thus (2.1) can be viewed as a problem in which nature, by choosing $P\epsilon\Pi$, determines the game to be played and hence, its payoff matrix $B(P)$, and then the players determine their optimal maximin mixed strategies for this game. The assumption underlying the maximum criterion of (2.1) is that, from the point of view of player I, nature will choose the most unfavorable payoff matrix form the continuum of payoff matrices $\{B(P),\ P\epsilon\Pi\}$. This is in contrast to (2.3) where nature was assumed to select the most favorable payoff matrix $B(P)$ from the point of view of player I.

To analyze (2.1), we first convert it to a more useful form by writing any $P\epsilon\Pi$ as a convex combination of the extreme points P_t, $t=1, 2, \ldots, T$ of the convex polyhedron Π.* Thus any $P\epsilon\Pi$ is of the form $P = \sum_{t=1}^{T} \lambda_t P_t$, where $\sum_{t=1}^{T} \lambda_t = 1 \quad \lambda_t \geq 0 \ \forall t$. Thus (2.1) becomes

$$\max_X \min_Y \min_\lambda \sum_{i=1}^{m} \sum_{j=1}^{n} x_i \left(\sum_{t=1}^{T} \left[\sum_{k=1}^{K} a_{ij}(k)p_t(k) \right] \lambda_t \right) y_j$$

subject to

$$\sum_{i=1}^{m} x_i = 1 \qquad x_i \geq 0 \qquad \forall_i$$

(2.4)
$$\sum_{j=1}^{n} y_j = 1 \qquad y_j \geq 0 \qquad \forall_j$$

$$\sum_{t=1}^{T} \lambda_t = 1 \qquad \lambda_t \geq 0 \qquad \forall_t,$$

where $p_t(k)$ is the kth element of the extreme point P_t of Π.

To solve (2.4) we make the change of variable $v_{jt} = y_j\lambda_t$, $j=1, 2, \ldots, n$, $t=1, 2, \ldots, T$. Then (2.4) becomes:

$$\max_X \min_V \sum_{i=1}^{m} x_i \left(\sum_{j=1}^{n} \sum_{t=1}^{T} \left[\sum_{k=1}^{K} a_{ij}(k)p_t(k) \right] v_{jt} \right)$$

*Note that this extreme point reduction is also used in the solution of two stage linear programming under uncertainty as in [5].

subject to

$$\sum_{i=1}^{m} x_i = 1 \qquad x_i \geq 0 \qquad \forall_i$$

(2.5)

$$\sum_{j=1}^{n} y_j = 1 \qquad y_j \geq 0 \qquad \forall_j$$

$$\sum_{t=1}^{T} \lambda_t = 1 \qquad \lambda_t \geq 0 \qquad \forall_t$$

$$v_{jt} = y_j \lambda_t \qquad \forall_{t,j}.$$

But from the constraints of (2.5) we have $v_{jt} \geq 0$ all j, t and

$$\sum_{j=1}^{n} \sum_{t=1}^{T} v_{jt} = \sum_{j=1}^{n} \sum_{t=1}^{T} y_j \lambda_t = \sum_{j=1}^{n} y_j \left(\sum_{t=1}^{T} \lambda_t \right) = \sum_{j=1}^{n} y_j = 1.$$

Hence for any given X in (2.5) an optimal V is of the form $V_{j't'} = 1$ and $v_{jt} = 0$, $j \neq j'$, $t \neq t'$ where the indices j', t' are such that

$$\sum_{i=1}^{m} x_i \sum_{k=1}^{K} a_{ij'}(k) p_{t'}(k) = \min_{(j,t)} \sum_{i=1}^{m} x_i \left(\sum_{k=1}^{K} a_{ij}(k) p_t(k) \right).$$

Moreover Y^* and λ^*, a corresponding optimal Y and λ are given by

$$Y^* = \begin{cases} 1 & j = j' \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{and } \lambda_t^* = \begin{cases} 1 & t = t' \\ 0 & \text{otherwise.} \end{cases}$$

Therefore (2.5) is equivalent to:

$$\max_{X} \min_{(j,t)} \sum_{i=1}^{m} x_i \left(\sum_{k=1}^{K} a_{ij}(k) p_t(k) \right)$$

subject to: $$\sum_{i=1}^{m} x_i = 1 \qquad x_i \geq 0 \qquad \forall i.$$

But this can be rewritten as:

$$\text{Max } \alpha$$

subject to:
$$\sum_{i=1}^{m} x_i \left(\sum_{k=1}^{K} a_{ij}(k) p_t(k) \right) \geq \alpha \qquad \forall j, t$$

(2.6)
$$\sum_{i=1}^{m} x_i = 1$$

$$x_i \geq 0 \qquad \forall i.$$

Since (2.6) is a linear programming problem it is easily solved for the optimal mixed strategy, X^*, for player I.

It is worthwhile to interpret the meaning of the variables v_{jt} in (2.5). We can regard (2.5) as a deterministic two person sum game with $m \times nT$ payoff matrix $B = (B(P_1), \ldots, B(P_t), \ldots, B(P_T))$, where $B(P_t)$ is the $m \times n$ payoff matrix whose ijth element is $b_{ij}(P_t) = \sum_{k=1}^{K} a_{ij}(k) p_t(k)$. Then $v_{jt} = y_j \lambda_t$ represents the probability that the coalition of player II and nature will play the jth column of $B(P_t)$, while λ_j is the probability that nature selects the payoff matrix $B(P_t)$ and y_j is the probability that player II plays the jth column of some $m \times n$ payoff matrix. Moreover in solving (2.6) as a linear programming problem, we will obtain the optimal dual variables, and since (2.6) is the problem player I solves to determine his optimal mixed strategy in the two person game with payoff matrix B, these dual variables will give the optimal mixed strategy for the coalition of player II and nature. Thus the optimal solution of (2.6) will also yield an optimal set of v_{jt} for (2.5). However, these v_{jt} do not necessarily determine a set of corresponding y_j or λ_t, nor is it an easy matter to get these y_j, λ_t even when the v_{jt} are determined. The determination of y_j and λ_t is, however, of no real concern to player I. All he wants to know is what effect the combined efforts of player II and nature will have on the game he plays, and not how this combined effort is divided up among player II and nature. Thus, for example, player I wants to know that the coalition will play the jth column of $B(P_t)$ say 10 percent of the time, but he does not care how much of this 10 percent is due to the actions of player II, or how much is due to nature.

The only difficulty in solving (2.6) arises in obtaining the elements $p_t(k)$ of the extreme points $P_1, P_2, \ldots, P_T$ of Π. To show how to do this, we shall introduce the terminology used in [4] and [6].

With each measure $P \epsilon \Pi$ let there be associated an admissible support $S(P)$ defined by

$$S(P) = \{k | p(k) > 0\}.$$

Let

$$S(\Pi) = \{S(P) | P \epsilon \Pi\}$$

be the set of admissible supports for Π and suppose that $S(\Pi)$ contains N different supports $S_1, S_2, \ldots, S_N$. Then for each of these supports we define a probability measure $P_l = (p_l(1), p_l(2), \ldots, p_l(K))$, $l = 1, 2, \ldots N$, such that

(2.7)
$$p_l(k) = \begin{cases} \dfrac{1}{|S_l|} & \text{for } k \epsilon S_l \\ 0, & \text{otherwise,} \end{cases}$$

where $|S_l|$ is the cardinality of S_l.

In Theorem (2.2) in [4] it was shown that the set of probability measures P_l, $l = 1, 2, \ldots, N$, includes the extreme points P_t, $t = 1, 2, \ldots, T$ of Π. Thus once $S(l)$ is known, the set of probability measures P_l, $l = 1, 2, \ldots, N$, given by (2.7) can be used in (2.6) in place of the extreme point measures P_t, $t = 1, 2, \ldots, T$. This substitution is permissible since any $P \epsilon \Pi$ can be expressed as a linear combination of any set of points which contains the extreme points. Thus the development which led to (2.6) is still valid when the set of extreme point measures is augmented by the additional $P \epsilon \Pi$ which are in the set of measures given by (2.7). Thus the problem which player I must solve to determine his optimal mixed strategy is:

$$\max \alpha$$

subject to

$$\sum_{i=1}^{m} x_i \left(\sum_{k=1}^{K} a_{ij}(k) P_l(k) \right) \geq \alpha \qquad \forall j, l$$

(2.8)

$$\sum_{i=1}^{m} x_i = 1$$

$$x_i \geq 0 \qquad \forall i.$$

Hence, once $S(\text{II})$ is known (2.8) can be solved. Methods for determining $S(\text{II})$ are illustrated in the examples below and discussed in [4] and [6].

EXAMPLE 2.1: Consider a 2×2 game with four possible state of nature and the following partial ordering:

$$p(1) \geq p(2) \geq p(4) \qquad \text{and } p(3) \geq p(4).$$

The admissible supports are

$$\{1\}, \quad \{3\}, \quad \{1, 3\}, \quad \{1, 2\}, \quad \{1, 2, 3\}, \quad \{1, 2, 3, 4\},$$

so that $N = 6$. These six sets are the only sets of states such that a probability measure P which has $p(k) > 0$ for all members of the set and $p(k) = 0$, otherwise is in Π. The following table gives the state values for the 2×2 matrices $A(k)$ $k = 1, 2, 3, 4$.

State k	1	2	3	4
True probability	5/12	4/12	2/12	1/12
$A(k)$	$\begin{pmatrix} 3 & 3 \\ 2 & 2 \end{pmatrix}$	$\begin{pmatrix} 3 & 1 \\ 4 & 4 \end{pmatrix}$	$\begin{pmatrix} 4 & 1 \\ 1 & 3 \end{pmatrix}$	$\begin{pmatrix} 2 & 3 \\ 3 & 3 \end{pmatrix}$

The actual expected value for each a_{ij} is $E(a_{11})=3.08, E(a_{12})=2.00, E(a_{21})=2.58, E(a_{22})=2.91$. The true probabilities are, of course, not known to the players. It is straightforward to verify that

$$\{1\}, \quad \{3\}, \quad \{1, 2\}, \quad \{1, 2, 3, 4\}$$

are the admissible supports which yield the extreme probability measures so that $T=4$.

Thus the extreme probability measures P_1, P_2, P_3, P_4 are as follows:

$$P_1=(1, 0, 0, 0)$$

$$P_2=(0, 0, 1, 0)$$

$$P_3=(1/2, 1/2, 0, 0)$$

$$P_4=(.25, .25, .25, .25).$$

Using this we calculate that

$$B(P_1)=\begin{pmatrix}3 & 3\\ 2 & 2\end{pmatrix} \quad B(P_2)=\begin{pmatrix}4 & 1\\ 1 & 3\end{pmatrix} \quad B(P_3)=\begin{pmatrix}3 & 2\\ 3 & 3\end{pmatrix}$$

$$B(P_4)=\begin{pmatrix}3 & 2\\ 2.5 & 3\end{pmatrix}.$$

Using model (2.6) means player I plays the game with payoff matrix

$$B=\begin{pmatrix}3 & 3 & 4 & 1 & 3 & 2 & 3 & 2\\ 2 & 2 & 1 & 3 & 3 & 3 & 2.5 & 3\end{pmatrix}.$$

By solving the linear program corresponding to the game β, we obtain the optimal strategy $x_1=0.4$, $x_2=0.6$, $\alpha=2.2$.* Thus player I will, on the average, receive a payoff of 2.2 on each play of the game. From the optimal dual variables to (2.6) we get the result that the optimal coalition strategy for player II and nature is $v_{11}=0.25$, $v_{12}=0.25$ and $v_{22}=0.5$. That is, the coalition will play the second column of $B(P_1)$ and the first column of $B(P_2)$ with probability 0.25 and the second column of $B(P_2)$ with probability 0.5.

If instead of (2.6) we solved (2.8) we would have to compute $B(P_5)=\begin{pmatrix}3.5 & 2\\ 1.5 & 1.5\end{pmatrix}$ corresponding to $P_5=(0.5, 0, 0.5, 0)$ and $B(P_6)=\begin{pmatrix}2.33 & 1.67\\ 2.33 & 3\end{pmatrix}$ corresponding to $P_6=(0.33, 0.33, 0.33, 0)$. It is easily verified that the optimal solution to this problem coincides with the optimal solution to (2.6).

* It should be noted that if the players had complete information the value would be 2.7.

We now change the optimality criterion from an expected value criterion to one in which player I wants to maximize the probability of his getting a payoff β. The problem, that player I wants to solve is obtained from (2.1) by replacing

$$a_{ij}(k) \text{ by } \phi_{ijk}(\beta) = \begin{cases} 1 & \text{if } a_{ij}(k) \geq \beta \\ 0 & \text{if otherwise.} \end{cases}$$

Following the same procedure which transformed (2.4) to (2.6) we see that player I's problem becomes the program:

$$\max \ \alpha$$

subject to

$$\sum_{i=1}^{m} x_i \left(\sum_{k=1}^{K} p_t(k) \phi_{ijk}(\beta) \right) \geq \alpha \qquad \forall j, t$$

(2.9)

$$\sum_{i=1}^{m} x_i = 1$$

$$x_i \geq 0 \qquad \forall i.$$

EXAMPLE 2.2: We let the ordering of the states and the values of $A(k)$ be the same as in Example (2.1). Thus we have the same set of extreme probability measures P_1, P_2, P_3, and P_4.

Suppose that player I wants to achieve a payoff of $\beta = 3$ from the game. Then for $P_1 = (1, 0, 0, 0)$

$$P(a_{11} \geq 3) = \sum_{k=1}^{4} \phi_{11k}(3) p_1(k)$$
$$= 1 + 0 + 0 + 0 = 1.$$

Similarly

$$P(a_{12} \geq 3) = 1$$
$$P(a_{21} \geq 3) = 0$$
$$P(a_{22} \geq 3) = 0.$$

Therefore, using model (2.9) player I plays the game with payoff matrix

$$C = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0.5 & 0.75 & 0.5 \\ 0 & 0 & 0 & 1 & 0.5 & 0.5 & 0.5 & 0.75 \end{pmatrix}.$$

The optimal solution to this game is $x_1 = 0.5$, $x_2 = 0.5$, $\alpha = 0.5$. Thus the player has a .5 probability of achieving a payoff level of 3.

III. A PRIORI DISTRIBUTIONS OVER ADMISSIBLE PROBABILITY MEASURES

In this section, we consider the case in which the player's information about the distribution function $F(A)$ consists of the partial order relations defining II and, in addition, an a priori distribu-

tion function $G(P)$ defined over the set of admissible probability measures $P\epsilon\Pi$. Thus $\int_\Delta dG(P)$ gives the players' a priori probability that the actual distribution over the states of nature $A(k)$, $k=1, 2, \ldots, K$ of A is a distribution P in the set $\Delta\epsilon\Pi$.

We now present a model for determining player I's optimal mixed strategy $X=(x_1, x_2, \ldots, x_m)$. As before, we let β represent the payoff level which player I wants to achieve from the game. We let $P_P(a_{ij} \geq \beta)$ be the probability that the random variable $a_{ij} \geq \beta$ when the probability measure over the states of nature of A is $P\epsilon\Pi$. Then, using the definition of $\phi_{ijk}(\beta)$ given in the previous section, we have

$$P_P(a_{ij} \geq \beta) = \sum_{k=1}^{K} \phi_{ijk}(\beta)p(k).$$

But, since $P\epsilon\Pi$ is itself a random variable with distribution function $G(P)$, the expected probability that $a_{ij} \geq \beta$ is given by

$$P(a_{ij} \geq \beta) = \int_{P\epsilon\Pi} P_P(a_{ij} \geq \beta)dG(p)$$

$$= \int_{P\epsilon\Pi} \sum_{k=1}^{K} \phi_{ijk}(\beta)p(k)dG(P)$$

$$= \sum_{k=1}^{K} \phi_{ijk}(\beta) \int_{P\epsilon\Pi} p(k)dG(P).$$

Let

$$\bar{p}(k) = \int_{P\epsilon\Pi} p(k)dG(P), \tag{3.1}$$

and let $\bar{P} = (\bar{p}(1), \ldots, \bar{p}(K))$.

Then $$P(a_{ij} \geq \beta) = \sum_{k=1}^{K} \phi_{ijk}(\beta)\bar{p}(k).$$

Thus, if player I is given an a priori probability distribution $G(P)$ over $P\epsilon\Pi$, he can use this information to determine, via (3.1), an expected probability distribution $\bar{P}\epsilon\Pi$. Then, instead of a partial information random payoff game, player I can play the complete information random payoff game in which $\bar{P}$ is taken to be the known distribution over the states of nature of A. Thus, player I would determine his optimal mixed strategy by solving:

$$\max \ \alpha$$

subject to

$$\sum_{i=1}^{m} x_i \left(\sum_{k=1}^{K} \phi_{ijk}(\beta)\bar{p}(k) \right) \geq \alpha \qquad \forall j$$

$$\sum_{i=1}^{m} x_i = 1 \tag{3.2}$$

$$x_i \geq 0 \qquad \forall i.$$

There is, however, another model player I could use for determining his optimal mixed strategy. To develop this model we first note that since $P \epsilon \Pi$ is a random variable, the quantity

$$R_{ij}(P) = P_P(a_{ij} \geq \beta) \tag{3.3}$$

is also a random variable whose domain is $P \epsilon \Pi$ and whose range is [0, 1]. Thus we can regard $R_{ij}(P)$ as the ijth element of an $m \times n$ random matrix R. Moreover, since the distribution function of $R_{ij}(P)$ can be calculated from $G(P)$, the known distribution function of the random variable P, the distribution function $H(R)$ of the random matrix R is a known function. But if we use E_H to denote the expected value operator using the distribution function H, then

$$E_H(R_{ij}(P)) = \int_{P \epsilon \Pi} R_{ij}(P)\, dG(P)$$

$$= \int_{P \epsilon \Pi} P_P(a_{ij} \geq \beta)\, dG(P)$$

$$= \sum_{k=1}^{K} \phi_{ijk}(\beta)\, \bar{p}(k).$$

Thus we see that (3.2) is the problem player I would solve to determine his optimal mixed strategy in the expected value game with random payoff matrix R. However, as we showed in the introduction, models for the solution of random payoff games need not use only the expected value optimality criterion. Thus we now extend (3.2) to consider optimality criterion similar to those discussed in [2] and [3].

To do this we let β' be player I's desired payoff level in the complete information random payoff game with payoff matrix R. Then, using model (1.1), player I should determine his optimal mixed strategy by solving

$$\max \alpha$$

subject to

$$\sum_{i=1}^{m} x_i P_H(R_{ij}(P) \geq \beta') \geq \alpha \qquad \forall j$$

$$\sum_{i=1}^{m} x_i = 1 \tag{3.4}$$

$$x_i \geq 0 \qquad \forall i,$$

where we have written P_H to make it clear that the probability is computed using the known distribution function $H(R)$. But, using (3.3)

$$P_H(R_{ij}(P) \geq \beta) = P_H(P_P(a_{ij} \geq \beta) \geq \beta'),$$

so that (3.4) becomes

$$\max \alpha$$

subject to

$$\sum_{i=1}^{m} x_i P_H(P_P(a_{ij} \geq \beta) \geq \beta') \geq \alpha \qquad \forall j$$

(3.5)
$$\sum_{i=1}^{m} x_i = 1$$

$$x_i \geq 0 \qquad \forall i.$$

Thus, in (3.4) and (3.5), player I wants to maximize the probability, α, that the probability of achieving payoff level β is at least β'. This is significantly different from (3.2) in which the player wants to maximize the expected probability of his achieving at least β. The difference between (3.2) and (3.5) is also clear from their respective deterministic payoff matrices. In (3.2) the ijth element in the payoff matrix is

$$E_G(P_P(a_{ij} \geq \beta)) = E_H(R_{ij}(P)),$$

while in (3.5) the ijth element is

$$P_H(P_P(a_{ij} \geq \beta) \geq \beta') = P_H(R_{ij}(P) \geq \beta').$$

By regarding $R_{ij}(P)$ as the ijth entry in an $m \times n$ random payoff matrix R, we have, in effect, applied the techniques of [2] to a new type of random payoff game. That is, a two-level random payoff game in the following sense. The probability measure P gives a distribution over the states of nature in the original random payoff game matrix A and then the probability measure H determined by the a priori distribution G gives a distribution over the payoffs in the second level payoff matrix R. Thus we see that (3.5) enables us to interpret a random payoff game, with a given a priori distribution $G(P)$, as a random payoff game of a random payoff game. This interpretation enables us to apply to (3.5) the extensions of (1.1) discussed in [2] and [3].

One interesting extension of (3.5) involves the use of weight functions for either or both of the desired payoff levels β and β'. To introduce these weight functions we let $w(t)$ represent the a priori weight assigned to all payoff levels t in the interval $[\beta_1, \beta_2]$. Similarly, we let $u(r)$ be the a priori weight assigned to all payoff levels r in an interval $[\beta_1', \beta_2']$. Then we generalize (3.5) to the problem:

$$\max \alpha$$

subject to

$$\sum_{i=1}^{m} x_i \int_{\beta_1'}^{\beta_2'} P_H\left(\left(\int_{\beta_1}^{\beta_2} P_P(a_{ij} \geq t)\, w(t)\, dt\right) \geq r\right) u(r)\, dr \geq \alpha \qquad \forall j$$

(3.6)

$$\sum_{i=1}^{m} x_i = 1$$

$$x_i \geq 0 \qquad \forall i.$$

Four special cases of $w(t)$ and $u(r)$ which are of interest are the following:

(a) $$u(r) = \begin{cases} 1 & r = \beta' \\ 0 & \text{otherwise} \end{cases} \qquad w(t) = \begin{cases} 1 & t = \beta \\ 0 & \text{otherwise;} \end{cases}$$

(b) $$u(r) = \begin{cases} 1/(\beta_2' - \beta_1') & \beta_1' \leq r \leq \beta_2' \\ 0 & \text{otherwise} \end{cases} \qquad w(t) = \begin{cases} 1 & t = \boldsymbol{\beta} \\ 0 & \text{otherwise;} \end{cases}$$

(c) $$u(r) = \begin{cases} 1 & r = \beta' \\ 0 & \text{otherwise} \end{cases} \qquad w(t) = \begin{cases} 1/(\beta_2 - \beta_1) & \beta_1 \leq t \leq \beta_2 \\ 0 & \text{otherwise;} \end{cases}$$

(d) $$u(r) = \begin{cases} 1/(\beta_2' - \beta_1') & \beta_1' \leq r \leq \beta_2' \\ 0 & \text{otherwise} \end{cases} \qquad w(t) = \begin{cases} 1/(\beta_2 - \beta_1) & \boldsymbol{\beta_1} \leq t \leq \beta_2 \\ 0 & \text{otherwise.} \end{cases}$$

Case (a) reduces (3.6) to (3.5). In case (b) a choice of β_1' and β_2' such that the interval $[\beta_1', \beta_2']$ contains the domain of the random variables $R_{ij}(P)$ for all i, j (for example $\beta_1' = 0, \beta_2' = 1$) will transform (3.6) to (3.2). Thus in case (b) the two-level random payoff games actually reduce to one level games by assuming the actual distribution, $P \epsilon \Pi$, to be the expected distribution $\bar{P} \epsilon \Pi$.

On the other hand, in case (c), an appropriate choice of β_1, β_2 leads player I to solve the problem:

$$\max \alpha$$

subject to

$$\sum_{i=1}^{m} x_i P_H(E_P(a_{ij}) \geq (\beta_2 - \beta_1)\beta' + \beta_1) \geq \alpha \qquad \forall j$$

(3.7)

$$\sum_{i=1}^{m} = 1$$

$$x_i \geq 0 \qquad \forall i,$$

in which, for each $P \epsilon \Pi$, a_{ij} has been replaced by its mean and then H has been used to compute the probability that this mean exceeds $\beta'(\beta_2 - \beta_1) + \beta_1$. Again (3.7) is a one-level random payoff game.

Finally, in case (d), (3.6) becomes

$$\max \alpha$$

subject to

$$\sum_{i=1}^{m} x_i E_H\left(\frac{E_P(a_{ij}) - \beta_1}{(\beta_2 - \beta_1)}\right) \geq \alpha \qquad \forall j$$

(3.8) $$\sum_{i=1}^{m} x_i = 1$$

$$x_i \geq 0 \qquad \forall i.$$

Therefore case (d) reduces both the first and second level random payoff games to expected value games. Thus we see that (3.6) contains considerable flexibility in modelling player I's problem.

In order to illustrate this type of model, we solve the following examples which use case (a) and (b).

EXAMPLE 3.1: Consider a 2×2 game with two states and the following state values:

$$\begin{array}{lcc} \text{States } k & 1 & 2 \\ A(k) & \begin{pmatrix} 3 & 2 \\ 1 & 2 \end{pmatrix} & \begin{pmatrix} 2 & 1 \\ 2 & 4 \end{pmatrix}. \end{array}$$

Consider the case where

$$w(t) = \begin{cases} 1 & t = \beta \\ 0 & \text{otherwise} \end{cases} \qquad \text{and } u(r) = \begin{cases} 1/(\beta_2' - \beta_1) & \beta_1' \leq r \leq \beta_2' \\ 0 & \text{otherwise,} \end{cases}$$

in which

$$H(P) = \begin{cases} 1/T & \text{for } P = P_t, \quad t = 1 \ldots T \\ 0 & \text{otherwise} \end{cases}.$$

Thus the a priori distribution $H(P)$ places equal weight of $1/T$ on all T extreme points of Π and assigns 0 probability to all other points of Π. Then $\bar{P}$ defined by (3.1) becomes the center of gravity of Π.

We use the same data as example (2.1). If we assume that player I wants to achieve a payoff level of $\beta = 3$, then using (3.1) we obtain:

$$P(a_{11} \geq 3) = 0.93$$

$$P(a_{12} \geq 3) = 0.5$$

$$P(a_{21} \geq 3) = 0.25$$

$$P(a_{22} \geq 3) = 0.56.$$

Then from (3.2) we have the optimal solution $x_1 = 0.42$, $x_2 = 0.58$, and $\alpha = 0.54$. Thus player I has a 54 percent probability of achieving a payoff level of 3.

EXAMPLE 3.2: Suppose that $\beta = 2$ and $\beta' = 0.6$ and $w(t)$ $u(r)$ are

$$w(t) = \begin{cases} 1 & t = \beta \\ 0 & \text{otherwise} \end{cases}$$

and

$$u(r) = t\begin{cases} 1 & r = \beta' \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$P(a_{11} \geq 2) = 1,$$

$$P(a_{12} \geq 2) = p, \qquad \text{where} \quad p = p(1)$$

$$P(a_{21} \geq 2) = 1 - p, \qquad \text{where } 1 - p = p(2),$$

$$P(a_{22} \geq 2) = 1.$$

Let $\Pi = \{p(1), p(2) : p(1) + p(2) = 1,\ p(1),\ p(2) \geq 0\}$ so that the players are not given any partial order information. Let $\frac{dH(P)}{dP} = h(P) = 2p$ on the interval [0, 1], where $H(P)$ is a distribution on $p(1)$. Then we have:

$$P_H(P_P(a_{11} \geq 2) \geq 0.6) = 1$$

$$P_H(P_P(a_{12} \geq 2) \geq 0.6) = 0.64$$

$$P_H(P_P(a_{21} \geq 2) \geq 0.6) = 0.16$$

$$P_H(P_P(a_{22} \geq 2) \geq 0.6) = 1.$$

Then, using (3.5), the problem player I want to solve is:

$$\max \alpha$$

subject to

$$x_1 + 16x_2 \geq \alpha$$

$$0.64x_1 + \quad x_2 \geq \alpha$$

$$x_1 + \quad x_2 = 1$$

$$x_1 \geq \quad 0x_2 \geq 0.$$

The optimal solution is $x_1 = 0.7$, $x_2 = 0.3$ and $\alpha = 0.748$. Thus player I has 75-percent confidence of achieving a payoff of 2 with probability .60.

COMMENTS

The previous sections suggest a number of future directions for research involving partial information. In the setting of (2.1), the problem of the value of information arises naturally. Specifically, if player I has the option of buy information about the distribution p, say in the form of reducing the

uncertainty set Π, at what point does the cost of the information outweigh the gains which result from the additional knowledge. In repeated plays of the game, such considerations lead to problems involving dynamic programming. If the information is statistical in nature, the determination of the optimal level of information about p seems difficult.

The value of the results of section 2 lies in the fact that they enable a player to determine optimal strategies when faced with partial information about the distribution. Such strategies will be robust in the sense that they do well against a whole family of distributions. Of course, the price paid for this robustness is that the guaranteed payoff may be lower than in the full information case. In this context, it seems reasonable to ask whether or not a player can determine a strategy which takes advantage of the fact that his opponent may only have partial information about the distribution of the payoffs. Up to now, we have been unable to make any significant progress on this problem.

BIBLIOGRAPHY

[1] Banös, A., "On Pseudo Games," Ann. of Math. Stat. *39*, 1932–1945 (1968).

[2] Cassidy, R. G., C. A. Field, and M. J. L. Kirby, "A Satisficing Model for Games with Random Payoffs," to appear in Management Science.

[3] Cassidy, R. G., C. A. Field, and M. J. L. Kirby, "Random Payoff Games II: A Payoff Maximization Model;" Working paper Urban Systems Institute, Carnegie-Mellon University (1970).

[4] Cassidy, R. G., C. A. Field, and M. J. L. Kirby, "Random Payoff Games with Partial Information: One Person Games Against Nature;" Revue Francaise d'Informatique et de Recherch Operationnelle, *3*, 3–18 (1971).

[5] Charnes, A., W. W. Cooper, and G. L. Thompson, "Constrained Generalized Medians and Linear Programming Under Uncertainty," Management Science *12*, 83–112 (1965).

[6] Fourgeaud, M. M., Lenclud, and Sentis, "Critere de Choix en Avenir Partiellement Incertain," R.I.R.O., *2*, 9–20 (1968).

[7] Harsanyi, J. C., "Games with Incomplete Information Played by 'Bayesian' Players, I–III, Part III. The Basic Probability Distribution of the Game," Management Science *14*, 486–502 (1968).

[8] Harsanyi, J. C., "Games with Incomplete Information Played by 'Bayesian' Players, I–III, Part II, The Basic Probability Distribution of the Game," Management Science *14*, 320–334 (1968).

[9] Harsanyi, J. C., "Games with Incomplete Information Played by 'Bayesian' Players, I–III, Part I, The Basic Model," Management Science *14*, 159–182 (1967).

ON SEQUENCING WITH EARLIEST STARTS AND DUE DATES WITH APPLICATION TO COMPUTING BOUNDS FOR THE $(n/m/G/F_{max})$ PROBLEM

Paul Bratley, Michael Florian, and Pierre Robillard

Département d'informatique
Université de Montréal

ABSTRACT

Recent efforts to improve lower bounds in implicit enumeration algorithms for the general $(n/m/G/F_{max})$ sequencing problem have been directed to the solution of an auxiliary single machine problem that results from the relaxation of some of the interference constraints. We develop an algorithm that obtains optimal and near optimal solutions for this relaxed problem with relatively little computational effort. We report on computational results achieved when this method is used to obtain lower bounds for the general problem. Finally, we show the equivalence of this problem to a single machine sequencing problem with earliest start and due date constraints where the objective is to minimize the maximum lateness.

1. INTRODUCTION

Recent efforts to improve the efficiency of implicit enumeration algorithms [1], [2], [4], and [7] for the general $(n/m/G/F_{max})$ job shop problem have been directed to the solution of an auxiliary single machine (resource) problem that results from the relaxation of some of the interference constraints as follows. Suppose that a partial schedule has been constructed. Then the earliest start of the set of operations in the set of "*schedulable*" *operations* (see [3], p. 113) is constrained by the earliest finish times of the operations that have been scheduled. A strong lower bound may be obtained by solving the one machine problem, for each of the machines that must still process some jobs, that is obtained when the interference constraints are relaxed for all machines except the one considered.

This one machine problem consists of solving a sequencing problem ot n or less jobs, where each job i is available for scheduling at time a_i, $a_i \geq 0$ (earliest start constraint), requires a processing time d_i on the machine and has an additional processing time q_i which is equal to the sum of its processing times on the remaining machines. The processing times q_i do not interfere with each other. The aim is to determine the order that minimizes the completion time for all jobs, including the q_i's. In the following we shall refer to this problem as the *relaxed problem*, as in [9].

Several attempts to compute good lower bounds for the relaxed problem have resulted in improved lower bounds for the general problem, and consequently reduced solution times. Schrage [8] obtained a lower bound of the relaxed problem by relaxing in addition the earliest start constraints. The optimal sequence for this problem is constructed by scheduling the jobs in decreasing order of the q_i's. A lower bound for the relaxed problem, and hence for the general problem, is the length of the largest resulting schedule plus $\min_i \{a_i\}$. In Ref. [4] the approach taken is to disregard the q_i's. An optimal sequence is obtained by scheduling jobs in increasing order of the a_i's. The corresponding lower bound

is the length of the largest schedule plus $\min_i \{q_i\}$. In Ref. [6] the reader may refer to computational results obtained with this bound. In both approaches the maximum over the machines of the one machine bound is a valid lower bound. Recently Schrage [9] partially combined the above two bounds to obtain an even stronger bound. We shall discuss his approach in detail in the following.

The purpose of this paper is to present an implicit enumeration algorithm for the relaxed problem which exploits its special structure. Optimal solutions are obtained with relatively little computational effort. We report on the use of this method to compute lower bounds for the general problem. The results that we obtained are the best reported to date in the literature. Finally, we show that this problem is equivalent to a single machine sequencing problem with earliest start and due date constraints where the objective is to minimize the maximum lateness of a finite number n of jobs.

2. THE ALGORITHM FOR THE RELAXED PROBLEM

Our approach in solving the relaxed problem consists of generating a good initial solution by using the method proposed by Schrage in [9]. We test whether this solution is optimal and if the test fails we proceed to construct new solutions in the framework of an implicit enumeration algorithm. Numerous exclusion rules, a strong lower bound and special branching rules, that are suggested by the structure of the relaxed problem, curtail the number of solutions generated explicitly.

It is first convenient to introduce some terminology. We shall refer to the value of d_i for a job as its duration, and the value of q_i as its tail. The critical job in a schedule will be that job whose scheduled starting time plus its duration plus its tail equals the total length of the schedule. If there are several such jobs, the critical job is the one scheduled earliest. A block is a group of jobs such that the first job is scheduled at its earliest start time, and all the remaining jobs to the end of the block are processed without any delays, but they do not start at their earliest start times. We say that there is a gap in the schedule whenever the machine is idle between jobs, even though the tails of previous jobs may not be finished. As a schedule is constructed we shall use S to denote the set of jobs so far in the schedule, and $\bar{S}$ to denote the remaining jobs.

The following three numerical examples of the problem will be used to illustrate various features of the solution method.

	Example 1				Example 2				Example 3		
i	a_i	d_i	q_i	i	a_i	d_i	q_i	i	a_i	d_i	q_i
1	3	6	9	1	1	8	3	1	6	5	4
2	4	4	17	2	8	4	12	2	7	1	6
3	14	7	9	3	10	7	4	3	10	3	10
4	15	2	7	4	14	3	7	4	10	9	8
5	17	5	8	5	16	5	18	5	19	2	5

We shall use the notation $i : j - k - l$ to mean that job i starts at time j, comes off the machine at time k, and is finished completely at time l. Also we will refer to k as b_i to relate it to job i.

Lower Bounds For The Relaxed Problem

Two lower bounds for the relaxed problem will be used in the following. The first bound is calculated very simply:

$$lb_1 = \max_i \{a_i + d_i + q_i\}.$$

The second bound is an improvement over that presented in [6] and requires somewhat more explanation. First, as in [6], the jobs are scheduled on the machine in order of ascending a_i. This will, in general, give a schedule made up of several blocks $B_1 \ldots B_k$. The lower bound is calculated by looking at the last block, B_k, and evaluating

$$lb_2 = \min_{i \epsilon B_k} a_i + \sum_{i \epsilon B_k} d_i + \min_{i \epsilon B_k} q_i.$$

In other words, we calculate a bound by assuming that the last job of all to be processed is the one with the shortest tail in the block.

One could obtain an even better bound by calculating a bound like lb_2 for each block in the schedule, not merely for the last one, and then maximizing over the blocks. (However, for simplicity, we have chosen not to do this in the computer program described later.)

Finally, we maximize over lb_1 and lb_2:

$$lb = \max (lb_1, lb_2).$$

For instance, in example 1 above, if the jobs are scheduled in order of increasing a_i, the schedule is

$$\begin{array}{l} 1:3 \ -9 \ -18 \\ 2:9 \ -13 -30 \\ 3:14 -21 -30 \\ 4:21 -23 -30 \\ 5:23 -28 -36 \end{array}$$

lb_1 is easily calculated to be 30. To calculate lb_2 we note that the schedule contains two blocks, $B_1 = \{1, 2\}$ and $B_2 = \{3, 4, 5\}$ and obtain $lb_2 = 14 + (7 + 2 + 5) + 7 = 35$. Thus the lower bound for example 1 is 35.

Similarly for example 2, $lb_1 = 39$ and $lb_2 = 32$ (there is only one block in the schedule), so a lower bound for example 2 is 39. Finally we may calculate the lower bound of 30 for example 3.

AN INITIAL SOLUTION OF THE RELAXED PROBLEM

An initial solution is constructed by the following algorithm, suggested by Schrage [9].

1. Set $t = -1$.
2. Is there at least one job $i \epsilon \bar{S}$ such that $a_i \leq t$? If so, go to 4.
3. Set $t = \min_{i \epsilon \bar{S}} (a_i)$.
4. Look at all the jobs $i \epsilon \bar{S}$, such that $a_i \leq t$. Choose that job, j say, whose value of q is a maximum. Break ties on the value of q by choosing the job with the higher value of d. If two available jobs have the same values of q and of d, choose either.
5. Add the chosen job to the schedule, and set $t = t + d_j$.
6. If $\bar{S} \neq \phi$, go to 2. Otherwise the schedule is complete, and its length may be determined by inspection.

For instance, the initial solution to example 1 is constructed as follows:

(a) $t = -1$.

(b) t becomes 3. Job 1 is the only one available, so it is selected. t becomes 9.

(c) Job 2 is the only one available, so it is selected. t becomes 13.

(d) t becomes 14. Job 3 is selected. t becomes 21.

(e) Jobs 4 and 5 are both available, but $q_5 > q_4$, so job 5 is selected. t becomes 26.

(f) Job 4 is selected, and the complete schedule is thus:

1: 3 − 9 − 18
2: 9 − 13 − 30
3: 14 − 21 − 30
5: 21 − 26 − 34
4: 26 − 28 − 35.

The length of the schedule is 35.

The reader may verify that the following are the initial solutions to examples 2 and 3:

Example 2	Example 3
1: 1 − 9 − 12	1: 6 − 11 − 15
2: 9 − 13 − 25	3: 11 − 14 − 24
3: 13 − 20 − 24	4: 14 − 23 − 31
5: 20 − 25 − 43	2: 23 − 24 − 30
4: 25 − 28 − 35	5: 24 − 26 − 31
Length = 43	Length = 31

The order of the jobs is the same as in the procedure suggested by Schrage [9], except that here we will work with the actual schedule duration obtained, whereas in [9] the largest $b_j + q_j$, for which q_j is no greater than the q_i's for jobs i scheduled before j in j's block, is used to obtain a lower bound for the relaxed problem.

Increasing the Lower Bound

We make the following assertion:

THEOREM 1: If the last scheduled job in the initial solution does not have the shortest tail in its block (or one of the shortest) then the lower bound lb_2 calculate above may be replaced by $lb_2 + 1$. (We assume that the processing times, d_i, are integers.)

PROOF: It is easy to see that the block structure of the initial solution and the block structure found when calculating lb_2 are the same, although the jobs within a block may be in a different order. In particular, the jobs making up the last block will be the same. We consider the initial scheduling of the jobs in this block.

If the last job of the last block does not have the shortest tail in the block, the job with the shortest tail (or the last remaining shortest-tailed job, if there are several) must obviously have been scheduled earlier. Consider the moment in time when we were about to add this job to the schedule. At that moment we had no choice, for:

(i) if a job with a longer tail had been available, we would have scheduled that instead (by rule 4); and

(ii) we had already delayed this critical moment as much as possible by choosing other jobs with

the same tail but a longer duration in preference to this one (again by rule 4).
Therefore it is clear that we are forced either to have as the last job in the block one whose tail is not the minimum length, or to leave a gap in the block so that other jobs become available. In either case, the lower bound lb_2 cannot possibly be attained, so we may replace it by lb_2+1. Q.E.D.

Hence if the above condition is true, and if $lb_2 \geq lb_1$, we have an improved lower bound for the problem.

The effect on the examples we have already seen is as follows: in example 1, the last tail in the schedule ($q_4=7$) is the shortest in its block (jobs 3, 4, and 5), so the condition does not apply; in example 2, the last tail ($q_4=7$) is not the shortest in its block (which consists of all the jobs), so lb_2 becomes 33 instead of 32, but lb_1 is still greater at 39 so no overall improvement is made; in example 3, the last tail ($q_5 = 5$) is again not the shortest in its block (all the jobs), so lb_2 becomes 31 instead of 30, as does the lower bound for the problem.

Optimality of the Initial Solution

THEOREM 2*: If the last scheduled job in any block of the initial solution has the shortest tail in its block (or one of the shortest) and it is the critical job, the initial solution is optimal.

PROOF: As in the proof of Theorem 1, we use the fact that the block structure of the initial solution and that of the solution that gives lb_2 are identical. Thus if the critical job is the one with the shortest tail in the block, the solution may not be improved, since interchanging its position with any other job in its block results in a schedule time that is at least as long. The equality occurs if there is another job in the block with a tail of the same length. The argument applies to any block that contains such a job. Q.E.D.

Having constructed the initial solution, if the optimality test fails, we check to see whether it is optimal by comparing its value to the lower bound. If they are the same, the solution is optimal; otherwise we augment lb_2 if possible and begin a search for better solutions using an implicit enumeration technique.

In the examples we have been following, the initial solution is optimal for examples 1 and 3, but it is not optimal for example 2.

Implicit Enumeration Scheme

From the initial node of the tree, which we call level 0, we branch to n new nodes on level 1. Each of these nodes represents the assignment of job i, $i=1, 2 \ldots n$, to be first in the schedule. Next we branch from each node on level 1 to $(n-1)$ nodes on level 2. Each of these nodes represents the assignment of one of the $(n-1)$ unassigned tasks to be second in the schedule. We continue in this fashion. In general on level k, $1 \leq k \leq n$, there are $(n-k+1)$ new nodes generated from each node on the preceding level. It is evident that all the $n!$ possible orderings of the jobs are enumerated in this way.

We now explain some properties of the relaxed problem that are used to cut down this tree-search by excluding from consideration certain parts of the tree.

(i) Recognition of an optimal solution: if a feasible solution is found whose value is equal to the calculated lower bound, the search is clearly at an end.

---

*This is the result implied in the bound proposal by Schrage in [9].

(ii) Recognition of a sufficiently good solution: in the application with which we are most concerned, we may not need to find an optimal solution, but simply one whose value does not exceed a predetermined limit. Again, if such a solution is found, the search need not continue. (See section 4.)

(iii) Exclusion: suppose we have just added job j to the schedule in some position. If for any $i \epsilon \bar{S}$ we find that $b_j + d_i + q_i$ equals or exceeds the best solution value so far obtained, we may exclude all the descendants of the node representing job j from further consideration.

(PROOF: the schedule will be at least as long as the best known schedule if job i immediately follows job j, and it clearly cannot be shortened by holding up job i till later. Q.E.D.)

(iv) Exclusion: suppose again that we have just added job j to the schedule, and let $i \epsilon \bar{S}$. We may exclude i from consideration as the immediate successor of j if there exists a $k \epsilon \bar{S}$ such that

$$\max (b_j, a_k) + d_k \leq a_i.$$

(PROOF: It is clear that in this case k cannot delay the start of i, so nothing can be lost, and something may be gained, by scheduling it before i. Q.E.D.)

(v) Forced choice: suppose we have just added job j to the schedule and $b_j \geq a_i$ for all $i \epsilon \bar{S}$. Then we may schedule the remaining jobs optimally simply by ordering them in decreasing order of q, excluding all other orderings of this subsequence from consideration.

(PROOF: We have a subproblem where all earliest start times are zero. The optimal order is then in decreasing order of the q_i's [8]. Q.E.D.)

(vi) Bounding: whenever we add a job to the schedule, we may recompute a lower bound for its finish time by a technique similar to the one previously described. If this lower bound equals or exceeds the previous best solution value, there is no point in trying to complete the schedule. We have chosen to recompute a bound whenever a gap is left in the schedule, but not otherwise.

(vii) Backup after the initial solution: by the method of construction of the initial solution any job started after time a_c and before job c (the critical job) must have a longer tail than c. There is therefore no need to consider other orderings of these jobs, and we may back up immediately to the level of that job j which is the first in the schedule to have $b_j \geq a_c$.

(viii) Backup after other solutions: to improve any solution the critical job must be moved forward. When a new solution is found we may therefore back up immediately to the level of the job immediately preceding the critical job in the schedule.

(ix) Problem decomposition: suppose we have just added job j to the schedule at level k, and $b_j \leq a_i$ for all $i \epsilon \bar{S}$. In this case we need never backtrack beyond level $k+1$ unless it is in consequence of rules (vii) or (viii) above.

(PROOF: an attempt to backtrack not made in consequence of rule (vii) or (viii) indicates that all possible orderings of the remaining $(n-k)$ jobs have been tried, implicitly or explicitly, and that the critical job was always one of these. But the best schedule for these jobs may not be started prior to the smallest earliest start time among these jobs. Thus a change in the order of the first k jobs cannot diminish the total time required for completion of the schedule. Q.E.D.)

(x) A heuristic: at level k we have in general $(n-k+1)$ nodes to consider. By analogy with rule (v), where the choice is optimal, we consider first a job $i \epsilon \bar{S}$, if any, for which $a_i \leq b_j$, where j is the last job added to the schedule, and (q_i, d_i) lexicographically greater than (q_l, d_l) for all $l \epsilon \bar{S}$ with $a_l \leq b_j$. Thereafter, or if no such job i exists, we consider the remaining jobs in increasing order of a_i's.

A Fortran code of the algorithm, implementing all the techniques mentioned above, is available from the authors.

We complete example 2 by displaying the tree search carried out to find the optimal solution. The tree generated is shown in Figure 1.

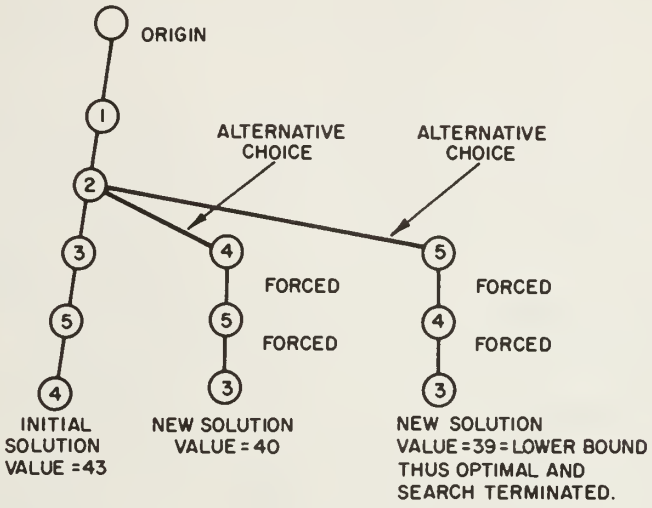


FIGURE 1.

3. TIMING TESTS OF THE ALGORITHM FOR THE RELAXED PROBLEM

The algorithm described above was tested on a series of randomly generated problems. The test problems were generated by choosing the number of jobs to be considered, n, and also three parameters $a_{\max}$, $d_{\max}$, and $q_{\max}$. Then the values of a_i, $i=1, 2, \ldots n$ were selected randomly from a uniform distribution between 1 and $a_{\max}$, the values of d_i were selected randomly from a uniform distribution between 1 and $d_{\max}$, and the values of q_i were selected randomly from a uniform distribution between 1 and $q_{\max}$.

The main results are presented in Table 1. It will be seen there that tests were carried out on 20-job problems and on 50-job problems. For a given problem size, the tests are divided into two series: the first series has $a_{\max} \doteq \frac{1}{2}nd_{\max}$, so that we would expect the resulting schedule on the machine to have few gaps in it, while the second series has $a_{\max} > \frac{1}{2}nd_{\max}$, so that we would expect the machine to have many idle periods. Within a series, tests were split into three kinds: the first kind have $q_{\max} < d_{\max}$, the second have $q_{\max} = d_{\max}$, and the third kind have $q_{\max} > d_{\max}$. One hundred tests of each kind were run, so that the table presents all told the results of 1200 tests.

TABLE 1

Number of jobs n	a_{max}	d_{max}	q_{max}	Failures to find optimum in 2 sec[a] out of 100	Remaining times:[a]		
					max	min	average
20	250	25	10	2	0.288	0.010	0.017
			25	19	0.348	0.010	0.019
			75	21	0.419	0.010	0.029
	1000	25	10	0	0.017	0.011	0.014
			25	0	0.018	0.012	0.014
			75	0	0.017	0.012	0.014
50	600	25	10	2	0.063	0.039	0.051
			25	7	0.212	0.040	0.051
			75	14	1.580	0.040	0.051
	2500	25	10	0	0.073	0.057	0.065
			25	0	0.070	0.055	0.063
			75	0	0.071	0.056	0.064

[a] All times are in seconds measured on a CDC 6400 using the optimizing Fortran compiler (FTN).

Problems where the optimum had not been found after about 2 seconds were stopped from running further, since it would evidently be impractical to allow times as long as this for solution. Each row of the table gives the number of occasions (out of 100) that this happened. For the remaining problems, the table gives the maximum, minimum, and average solution times. All times quoted are in seconds, measured on a CDC 6400 using the optimizing Fortran compiler. Thus, for instance, we see that for 20-job problems with $a_{max} = 250$, $d_{max} = 25$, and $q_{max} = 10$, the algorithm twice failed to find the optimum within 2 seconds, and in the remaining 98 tests the longest time taken was 0.228 seconds, the shortest was 0.010 seconds, and the average was 0.017 seconds.

The results in the table may be seen to support the following (not very startling) conclusion: difficulty is experienced in solving those problems in which there are few gaps in the schedule and in which the tails of the jobs are as long as or longer than their durations, on average; on the contrary, no difficulty is generally expected in solving problems in which there are many gaps in the schedule and the solution times for this kind of problem are remarkably constant.

Among the results not presented in the table, since no systematic tests were carried out, we may note the following: problems involving only two jobs are solved in about 1 millisecond; problems involving 10 jobs take about 6 milliseconds; and in one series of 16 trials with 80-job problems, the algorithm failed to find the optimum once, while the average of the other 15 trials was 0.111 second.

TABLE 2

Job network type	Size Job/Mach	Our New Bound				Bound Given in [6]			
		Average number of nodes examined		Average CDC 6400 time/prob (sec)	Number of problems solved out of 10	Average number of nodes examined		Average CDC 6400 time/prob (sec)	Number of problems solved out of 10
		To obtain the optimal solution	To prove optimality			To obtain the optimal and best [a] solution	To prove optimality		
Random	5/4	30	30	0.56	10	75	87	0.52	10
	7/4	110	125	8.82	10	[a] 380	[a] 637	7.70	8
Fixed random	5/4	25	25	0.30	10	[a] 56	[a] 436	3.13	8
	7/4	29	29	0.49	10	168	168	1.38	10
Permutation	5/4	22	26	0.29	10	36	72	0.40	10
	7/4	99	146	2.54	10	[a] 352	[a] 477	4.24	9
Flow-shop	5/4	103	122	1.52	10	105	178	1.18	10
	7/4	194	619	11.39	10	[b] 329	[b] 1163	12.98	8
Flow-shop [c]	5/4	75	83	0.94	10	83	112	0.62	10
	7/4	199	336	5.81	10	516	699	7.34	10

Processing times were cnosen from a uniform distribution of integers over [1, 10].

[a] Includes problems terminated at 2,000 iterations.

[b] Includes problems terminated at 4,000 iterations.

[c] All generated schedules have the same job order on machines 1 and 2 and the same job order on machines $m-1$ and m.

4. NUMERICAL RESULTS OBTAINED FOR THE GENERAL PROBLEM

We adapted the algorithm of section 2 to obtain lower bounds for the general problem in the usual way. Several slight modifications were effected in order to avoid unnecessary computations. If the initial solution for the relaxed problem was not optimal we limited the number of nodes generated in the implicit enumeration scheme to 1000. This arbitrary rule prevents excessive calculations when a "bad" relaxed problem is met. In addition, since we are interested in finding the maximum of the relaxed problems over the set of machines, we stopped calculations if the value of the initial solution (not necessarily optimal) was smaller than that of a solution obtained for a machine considered previously.

We tested this new bound against the bound of [6] using the implicit enumeration algorithm given in [4] on a series of 80 job shop problems of different type: random, fixed random, permutation, and flow shop (see [3]). The results are given in Table 2. These indicate that the new bound reduces the number of nodes examined drastically, however the solution times, although reduced, do not decrease in the same proportion. This is due to the larger computational effort spent in order to obtain optimal or near-optimal solutions for the relaxed problems.

We also solved the three scheduling problems given in [5]. These have become benchmark testing problems for algorithms that solve the general sequencing problem. The interesting fact to note about the results, given in Table 3, is that for problems 2 and 3 the value of the initial solution obtained with our bound is better than the best solution known previously. On the other hand, the failure to prove optimality for problems 2 and 3 in spite of the large number of nodes generated, with use of the "best possible bound," indicates that methods other than bounds must be used to further curtail the tree search in the general problem.

5. SCHEDULING WITH EARLIEST START AND DUE DATE CONSTRAINTS

The relaxed problem may be shown to be equivalent to a single machine sequencing problem with earliest start and due date constraints. In this problem, each job i is available for scheduling at time a_i, $a_i \geq 0$ (earliest start constraint), requires a processing time d_i on the machine and must be completed before time b_i (due date constraint). The objective is to minimize the maximum lateness of any job, where lateness is the time by which any job exceeds or preceeds its due date.

The equivalence is accomplished by the following simple transformation. Select a completion time, T, that is sufficiently large to exceed $\max_i \{b_i\}$, and define additional processing times q_i as

$$q_i = T - b_i.$$

TABLE 3

Problem number	Number of machines	Number of operations	Value of previously best computed solution	Value of the first solution	Value of the best obtained	Number of iterations to optimal and best [a]	Number of iterations to prove optimal	CDC 6400 time (sec)
1	6	36	55	55	55	36	59	1.52
2	5	100	1201	[b] 1198	1175	[a] 1500	[a] 1542	939.87
3	10	100	1041	[c] 1037	980	[a] 5908	[a] 17278	939.56

[a] Indicates that calculations were terminated without proving optimality. For the problems (2) and (3) the best unexamined bounds were 1164 and 808, respectively.

[b] The time spent to find the first solution was 105.03 sec.

[c] The time spent to find the first solution was 19.87 sec.

We have now a relaxed problem with earliest starts, a_i, processing times, d_i, and tails, $(T - b_i)$. We solve this problem by our algorithm. Let the optimal completion time be T^*. $T^* - T$ is the value of the min-max lateness, and if $T^* \leq T$, then all jobs are processed before their due date.

ACKNOWLEDGMENT

We are grateful to Professor Linus Schrage for numerous relevant remarks on an earlier version of this paper.

REFERENCES

[1] Balas, E., "Machine Sequencing via Disjunctive Graphs: An Implicit Enumeration Algorithm," Operations Research, Vol. 17 (Nov.–Dec. 1969).

[2] Brooks, G. H. and C. R. White, "An Algorithm for Finding Optimal or Near Optimal Solutions to the Production Scheduling Problem," J. Indust. Eng. (Jan.–Feb 1965).

[3] Conway, R. W., W. L. Maxwell, and L. W. Miller, *Theory of Scheduling* (Addison-Wesley, 1967).

[4] Florian, M., P. Trépant, and A. MacMahon, "An Implicit Enumeration Algorithm for the Machine Sequencing Problem," Management Science: Applications, *17*, pp. 782–792 (Aug. 1971).

[5] Muth, J. F. and G. L. Thompson, *Industrial Scheduling* (Prentice-Hall, 1963).

[6] Sang, Nguyen and M. Florian, "A Note on Lower Bounds for the Machine Sequencing Problem." Publication No. 49, Dépt. d'informatique, Université de Montréal (Dec. 1970).

[7] Schrage, L., "Solving Resource Constrained Network Problems by Implicit Enumeration—Nonpreemptive Case," Operations Research, Vol. 18, No. 2 (Mar.–Apr. 1970).

[8] Schrage, L., "A Bound Based on the Equivalence of Min. Max. Completion Time and Min. Max. Lateness Scheduling Objectives." Center for Mathematical Studies in Business and Economics Report, University of Chicago, Sept. 1970.

[9] Schrage, L., "Obtaining Optimal Solutions to Resource Constrained Network Scheduling Problems" (Mar. 1971).

SCHEDULING JOBS, WITH EXPONENTIALLY DISTRIBUTED PROCESSING TIMES, ON TWO MACHINES OF A FLOW SHOP

Andrew A. Cunningham

and

Sujit K. Dutta

University of Toronto
Toronto, Canada

ABSTRACT

This paper treats the problem of sequencing n jobs on two machines in a "flow shop." (That is, each job in the shop is required to flow through the same sequence of the machines.) The processing time of a given job on a given machine is assumed to be distributed exponentially, with a known mean. The objective is to minimize the expected job completion time. This paper proves an optimal ordering rule, previously conjectured by Talwar [10]. A formula is also derived through Markov Chain analysis, which evaluates the expected job completion time for any given sequence of the jobs. In addition, the performance of a heuristic rule is discussed in the light of the optimal solution.

INTRODUCTION

Problems of sequencing a given set of jobs on a given set of machines have been considered by many researchers. A review of some of these works, along with an extensive bibliography, are given in References 4 and 5. Most of these studies consider that the processing times (or associated costs) are fixed, and that they are known prior to making sequencing decisions.

In most practical situations, however, the time taken to process a job on a machine is a random variable. Such stochastic variations of the processing times make the sequencing problems far more complex than the corresponding deterministic ones. Banerjee [3] and Makino [8] are perhaps the pioneers in reporting this class of problems. Banerjee solved a single machine sequencing problem using a criterion of minimizing the maximum probability of lateness of any job. Makino considered the problems of sequencing two jobs on two machines, and two jobs on three machines when the processing times follow exponential distributions. The objective function was to minimize the expected "elapsed time" (i.e., the time elapsed from the start of the first job on the first machine until the completion of the last job on the last machine). After solving a three-job, two-machine problem with exponentially distributed processing times, Talwar [10] conjectured an optimal ordering rule for the general n-job, two-machine case. Bagga [1] extended Talwar's approach to consider two-job, three-machine problems when the processing times are generally distributed. He also considered a different objective function. In a later paper Bagga [2] extended the calculations performed by Talwar by proving the conjecture for the four-job, two-machine problem when the processing times follow exponential distributions. However, for the general n-job problem the criterion remains a conjecture.

STATEMENT AND SOLUTION OF PROBLEM

Let us define the problem through the following assumptions:

1. There are n jobs simultaneously available, and these are to be processed by each of two given machines.

2. The shop is a "flow shop." That is, each job in the shop is required to flow through the same sequence of machines.

3. The setup time for a given job on a given machine, if it exists, is independent of the sequence in which the jobs are processed.

4. Let A_i be the random processing time (including setup, if any) of the ith job on the first machine, and B_i the corresponding time on the second machine. A_i and B_i are independent and exponentially distributed with known parameters, a_i and b_i, respectively, $i = 1, 2, \ldots, n$.

5. No machine may process more than one job at a time. Also a job cannot be processed by more than one machine simultaneously.

6. All jobs are considered equal in importance. Thus there are no due dates, priorities, or rush orders.

7. No preemption is allowed (i.e., once a job is started on a machine it must be processed on the machine to completion).

8. In-process inventory (otherwise called buffer-stock) is allowed. (This ensures that there is no idle time on the first machine until it has processed all the jobs.)

9. The objective of the problem is to sequence the n jobs so as to minimize the expected elapsed time.

The following Lemma gives a necessary condition for an optimal solution.

LEMMA 1: In order to find the optimal schedule for the problem it is sufficient to consider only those schedules which have identical job ordering on both the machines.

PROOF: This result has already been proved for the deterministic equivalent of our problem (see [4], pp. 80, or [7]).

Let $\underset{\sim}{x} \equiv (A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_n)$, where A_i and B_i are as defined in the previous section. Let $p(\underset{\sim}{x})$ denote the probability density function of the vector random variable x. Now let u represent a joint sequence for processing the jobs successively on the machines; that is a sequence for the first machine followed by a sequence relating to the second machine. Then define the function $T_u(\underset{\sim}{x})$ to be the elapsed time for the sequence u with the processing times being the appropriate components of $\underset{\sim}{x}$.

Now consider two schedules s and s'. Schedule s has the jobs ordered identically on the two machines, while s' has the same ordering as s for the first machine, but a different ordering on the second. Therefore from the knowledge of the equivalent deterministic problem [4, 7], we will have for any realized value of the random vector $\underset{\sim}{x}$,

$$T_s(\underset{\sim}{x}) \leq T_{s'}(\underset{\sim}{x})$$

so that

$$\int_{\underset{\sim}{x}} T_s(\underset{\sim}{x}) \cdot p(\underset{\sim}{x})\, dx \leq \int_{\underset{\sim}{x}} T_{s'}(\underset{\sim}{x}) \cdot p(\underset{\sim}{x})\, dx, \text{ as } p(\underset{\sim}{x}) \geq \underset{\sim}{0}$$

or

$$E[T_s(x)] \leq E[T_{s'}(x)].$$

Thus the lemma is proved since the objective is to minimize the expected elapsed time.

Now we are in a position to prove Talwar's conjecture, and this is given in the following theorem.

THEOREM: An optimal sequence is determined by the criterion: job (j) precedes job $(j+1)$ if

$$(a_j + b_{j+1}) \geq (a_{j+1} + b_j). \tag{1}$$

PROOF: Let X_i denote the idle time of the second machine when waiting to start the ith job. (Note that X_i may be zero.) Identify the jobs by integers from one to n. Then the expected elapsed time for processing the jobs in the "natural" sequence, $1, 2, \ldots, j, j+1, \ldots, n$, is given by

$$T = E\left[\sum_{i=1}^{n} B_i + \sum_{i=1}^{n} X_i\right]$$

$$= E\left[\sum_{i=1}^{n} X_i\right] + \sum_{i=1}^{n} 1/b_i.$$

Thus the objective reduces to minimizing

$$T_1 = E\left[\sum_{i=1}^{n} X_i\right]$$

$$= E\left[\operatorname{Max}\left\{A_1, \operatorname*{Max}_{2 \leq j \leq n}\left[\left(\sum_{i=1}^{j} A_i - \sum_{i=1}^{j-1} B_i\right)\right]\right\}\right]$$

$$= E\left[\operatorname{Max}\left\{\operatorname*{Max}_{1 \leq k \leq j}\left(\sum_{i=k}^{j+1} B_i - \sum_{i=k+1}^{j+1} A_i\right), B_{j+1}, A_{j+2}, \operatorname*{Max}_{j+3 \leq k \leq n}\left(\sum_{i=j+2}^{k} A_i - \sum_{i=j+2}^{k-1} B_i\right)\right\}\right] + \sum_{i=1}^{j+1} (1/a_i) - (1/b_i)$$

$$= E\left[\operatorname{Max}\left(\xi_1, \eta\right)\right] + k,$$

where

$$\xi_1 = \operatorname{Max}\left\{\operatorname*{Max}_{1 \leq k \leq j}\left(\sum_{i=k}^{j+1} B_i - \sum_{i=k+1}^{j+1} A_i\right), B_{j+1}\right\} \tag{2}$$

$$\eta = \operatorname{Max}\left\{A_{j+2}, \operatorname*{Max}_{j+3 \leq k \leq n}\left(\sum_{i=j+2}^{k} A_i - \sum_{i=j+2}^{k-1} B_i\right)\right\}$$

$$k = \sum_{i=1}^{j+1} \{(1/a_i) - (1/b_i)\}.$$

If we interchange the jth and $(j+1)$st jobs in the above sequence, the expected idle time on the second machine becomes

$$T_2 = E\left[\operatorname{Max}\left(\xi_2, \eta\right)\right] + k,$$

where

$$\xi_2 = \text{Max}\left\{\underset{1\le k\le j-1}{\text{Max}}\left(\sum_{i=k}^{j+1} B_i - \sum_{i=k+1}^{j+1} A_i\right), B_j + B_{j+1} - A_j, B_j\right\}.$$

Note that η is independent of ξ_1 and ξ_2. Therefore

$$T_i = \int_0^\infty \lambda \cdot \frac{d}{d\lambda}\left[\left(\int_{-\infty}^{\lambda} f_i(\xi_i)\, d\xi_i\right)\left(\int_{-\infty}^{\lambda} g(\eta)\, d\eta\right)\right] d\lambda + k$$

where $f_i(\xi_i)$, $(i=1,2)$ and $g(\eta)$ are the p.d.f.s of ξ_i and η, respectively. Now, set

$$(3)\qquad S = T_1 - T_2 = \int_0^\infty \lambda \cdot \frac{d}{d\lambda}\left[\left\{\left(\int_{-\infty}^{\lambda} f_1(\xi_1)\, d\xi_1\right) - \left(\int_{-\infty}^{\lambda} f_2(\xi_2)\, d\xi_2\right)\right\}\left\{\int_{-\infty}^{\lambda} g(\eta)\, d\eta\right\}\right] d\lambda.$$

Note that with respect to the initial ordering, job j precedes job $j+1$ if $S \le 0$. Integrating (3) by parts we get

$$(4)\qquad S = \left[\lambda\{(F_1(\lambda) - F_2(\lambda))\, G(\lambda)\}\right]_0^\infty - \int_0^\infty \{F_1(\lambda) - F_2(\lambda)\} \cdot G(\lambda)\, d\lambda,$$

where F_i and G are, respectively, the cumulative distribution functions of ξ_i and η. As each of A_i and B_i $(i=1,2,\ldots,n)$ is exponentially distributed, the distribution functions of η and ξ_i $(i=1,2)$ will contain exponential terms. Hence in the limit, as $\lambda \to \infty$, these functions will approach unity at a faster rate than $1+(1/\lambda)$; as a result $\underset{\lambda\to\infty}{\text{Lim}}\, \lambda[F_1(\lambda) - F_2(\lambda)] = 0$. Thus

$$S = -\int_0^\infty \{F_1(\lambda) - F_2(\lambda)\} G(\lambda)\, d\lambda.$$

A sufficient condition for $S \le 0$ is

$$(5)\qquad F_1(\lambda) - F_2(\lambda) \ge 0, \forall \lambda \ge 0.$$

From (2) we can rewrite ξ_1 as follows:

$$\xi_1 = \text{Max}\left[\left\{\underset{1\le k\le j-2}{\text{Max}}\left(\sum_{i=k}^{j-1} B_i - \sum_{i=k+1}^{j-1} A_i\right), B_{j-1}\right\}, \{A_j, A_{j+1} + A_j - B_j\}\right] + B_j + B_{j+1} - A_j - A_{j+1}$$

or,

$$(6)\qquad \xi_1 = \text{Max}[X, Y_1] + Z$$

where

$$X = \text{Max}\left[\underset{1\le k\le j-2}{\text{Max}}\left(\sum_{i=k}^{j-1} B_i - \sum_{i=k+1}^{j-1} A_i\right), B_{j-1}\right],$$

(7) $$Y_1 = \text{Max}\,[A_j, A_{j+1} + A_j - B_j],$$

(8) $$Z = B_j + B_{j+1} - A_j - A_{j+1}.$$

It may be noted that X is independent of Y_1 and Z, but that Y_1 and Z are correlated. Also, X and Z are symmetric with respect to the interchange of the jth and $(j+1)$ jobs. Thus

$$F_i(\lambda) = Pr\{\text{Max}\,(X, Y_i) + Z \leq \lambda\},\ i = 1, 2.$$
$$\equiv F_{X+Z,\,Y_i+Z}(\lambda, \lambda).$$

As neither X nor Y_i takes negative values, we may write

$$F_i(\lambda) = \int_{\mu=-\infty}^{\lambda} F_X(\lambda - \mu) \cdot F_{Y_i|Z=\mu}(\lambda - \mu)\, f_Z(\mu)\, d\mu,$$

where $f_Z(\cdot)$ is the *p.d.f.* of Z.

Let $$P_i = F_{Y_i|Z=\mu}(\lambda - \mu) \cdot f_Z(\mu), \qquad i = 1, 2$$

(9) $$\equiv Pr\{Y_i \leq \lambda - \mu,\ \mu \leq Z \leq \mu + d\mu\}.$$

Then a sufficient condition for $F_1(\lambda) - F_2(\lambda) \geq 0$ is

(10) $$P_1 - P_2 \geq 0, \qquad \forall \lambda \geq 0, \qquad \mu \leq \lambda.$$

Substituting the values of Y_1 and Z from (7) and (8), respectively, into (9), we have

(11) $$P_1 = Pr\{A_j \leq \lambda - \mu, \qquad B_{j+1} \leq \lambda, \qquad \mu \leq B_j + B_{j+1} - A_j - A_{j+1} < \mu + d\mu\}$$

$$= \int_{u_j=0}^{\lambda-\mu} \int_{v_{j+1}=0}^{\lambda} f_{A_{j+1}-B_j}(v_{j+1} - \mu - u_j)\, f_{B_{j+1}}(v_{j+1}) \cdot f_{A_j}(u_j)\, dv_{j+1} \cdot du_j.$$

The appropriate density functions may now be substituted in (11) and after properly adjusting the limits the integrals may be easily evaluated. Thus we obtain the expression for P_1. Note that P_2 may be directly obtained from P_1 by interchanging a_j with a_{j+1} and b_j with b_{j+1}. Then we can find the difference $P_1 - P_2$, which after lengthy manipulations, turns out to be

$$P_1 - P_2 = \phi(\lambda - \mu) \cdot \Psi(\lambda)\,[(a_j + b_{j+1}) - (a_{j+1} + b_j)],$$

where

$$\phi(\lambda - \mu) = [\exp\{-a_j(\lambda - \mu)\} - \exp\{-a_{j+1}(\lambda - \mu)\}] \cdot a_j \cdot a_{j+1} \cdot b_j \cdot b_{j+1} / (a_j + b_{j+1})(a_{j+1} + b_j)$$

$$(a_{j+1} - a_j)$$

and

$$\Psi(\lambda) = [\exp(-b_{j+1} \cdot \lambda) - \exp(-b_j \cdot \lambda)]/(b_j - b_{j+1}).$$

It is interesting to notice that

$$\phi(\lambda - \mu) \geq 0, \qquad \forall(\lambda - \mu) \geq 0$$

and

$$\Psi(\lambda) \geq 0, \qquad \forall\lambda \geq 0.$$

Therefore,

(12) if $(a_j + b_{j+1}) \geq (a_{j+1} + b_j)$ then $P_1 - P_2 \geq 0, \forall\lambda > 0, \forall\mu \leq \lambda$

Combining (4), (5), (10), and (12), it may be concluded that job j should precede job $(j+1)$ if

$$(a_j + b_{j+1}) \geq (a_{j+1} + b_j).$$

(13) $$[\text{i.e., } (a_j - b_j) \geq (a_{j+1} - b_{j+1})]$$

LEMMA 2: Relation (1) is transitive.

The proof is a straightforward extension of that given in Reference 10, pp. 96.

DETERMINATION OF EXPECTED ELAPSED TIME

Having found the optimal job sequence one may often be interested in evaluating the expected elapsed time for that sequence. Prior to this work, however, no closed form expression was available giving the expected elapsed time for an n-job problem. Bagga [2] gave an expression involving n-fold integrals, but for these he gave no resulting solution. Moreover, since these integrals are interrelated, Bagga is in error in the statement, "It may be mentioned here that as the integrals are exponentials, the multiple integrals can easily be converted into products of single integrals and the calculation becomes easy," [2, pp. 195]. Thus it is not obvious how to obtain via his method a general expression for the expected elapsed time of an n-job problem.

This difficulty prompted us to investigate alternative approaches to the problem. We have found that the problem can be conveniently tackled with the help of Markov Chain analysis, as will now be described.

Consider n jobs being processed on two machines in the natural sequence s:1, 2, 3, . . ., j, $j+1$, . . ., n, as shown in Figure 1. This is clearly equivalent to a queuing system, where n already-arrived customers are to be served by two service channels in series, and where there is sufficient waiting room between the two service stations so that all customers waiting for the second channel can be accommodated.

Let (q, r) $(q, r = 1, 2, \ldots n+1;\ q \geq r)$ be a doublet representing the state of the flow shop, and let the state occupancy probabilities be defined as follows:

(14) $P_{q,r}(t)$ = Prob (job q is on machine 1 and job r is on machine 2 at time t), $n \geq q > r \geq 1$;
= Prob (job q is on machine 1 and machine 2 is waiting for job q at time t), $n \geq q = r \geq 1$;
= Prob (all jobs have been processed by machine 1 and job r is on machine 2 at time t), $q = n+1, n \geq r \geq 1$;
= Prob (all jobs have been processed by both machines at time t), $q = r = n+1$.

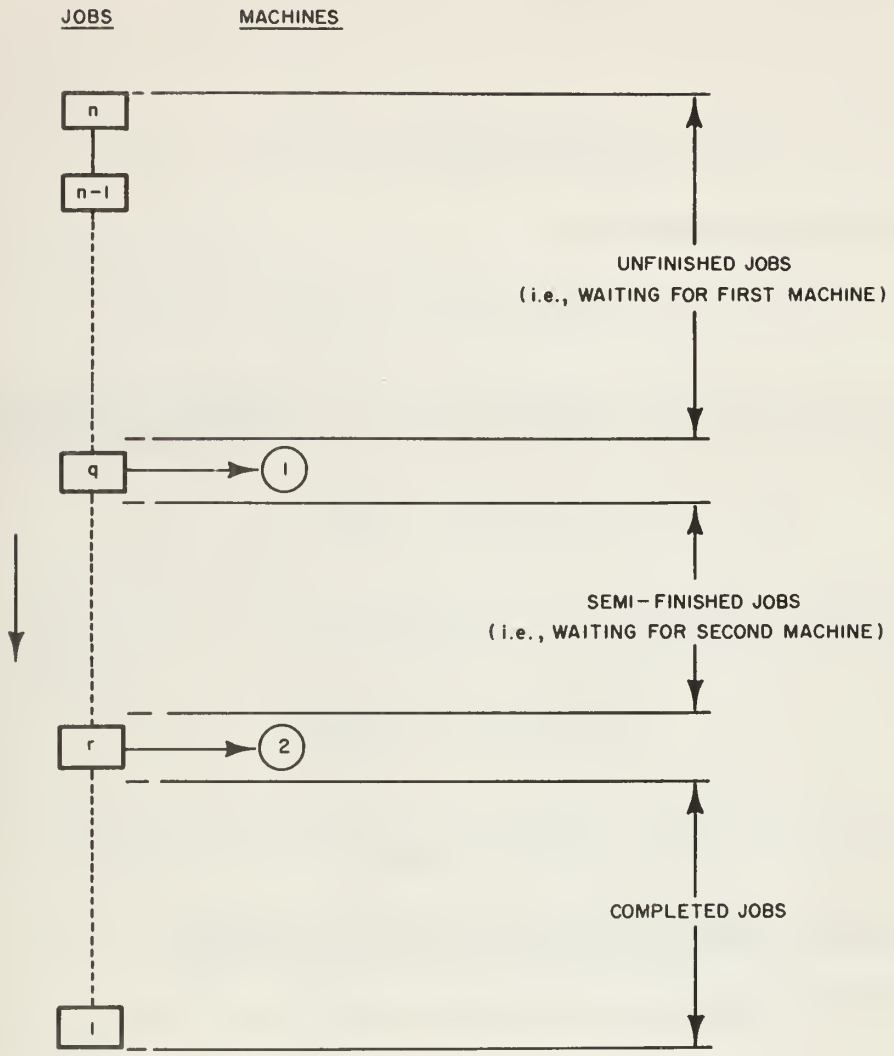


FIGURE 1. Schematic diagram of the production system

It may be seen that $\{(q, r), q, r=1, 2, \ldots, n+1; q \geq r\}$ is a time-homogeneous Markov Chain, since the mean processing rates, $1/a_i$ and $1/b_i$, $i=1, 2, \ldots, n$, are positive constants. The associated Chapman-Kolmogorov equations may be written as:

$$P_{q,r}(t+\Delta t) = P_{q,r}(t)\cdot[1-(a_q+b_r)\cdot\Delta t] + P_{q-1,r}(t)\cdot a_{q-1}\cdot\Delta t + P_{q,r-1}(t)\cdot b_{r-1}\cdot\Delta t, \tag{15}$$

$$n+1 \geq q \geq 1.$$

$$P_{q,q}(t+\Delta t) = P_{q,q}(t)\cdot[1-a_q\cdot\Delta t] + P_{q,q-1}(t)\cdot b_{q-1}\cdot\Delta t, \qquad n+1 \geq q \geq 1, \tag{16}$$

where $b_0 \equiv a_{n+1} \equiv 0$.

In the above two sets of equations, if we let $\Delta t \to 0$, then in the limit we get

$$P'_{q,r}(t) = -(a_q + b_r)\cdot P_{q,r}(t) + a_{q-1}\cdot P_{q-1,r}(t) + b_{r-1}\cdot P_{q,r-1}(t), \qquad n+1 \geq q > r \geq 1 \tag{17}$$

$$P'_{q,q}(t) = -a_q \cdot P_{q,q}(t) + b_{q-1}\cdot P_{q,q-1}(t), \qquad n+1 \geq q \geq 1. \tag{18}$$

And we have the following initial condition

$$\left.\begin{aligned} P_{q,r}(0) &= 1, && \text{if } q = r = 1 \\ &= 0, && \text{otherwise} \end{aligned}\right\}. \tag{19}$$

In order to solve the above set of differential equations we introduce the Laplace-Stieltjes Transform:

$$L\{P_{q,r}(t)\} \equiv P^*_{q,r}(\theta) = \int_0^\infty \exp(-\theta t) P_{q,r}(t)\, dt, \qquad (\theta > 0).$$

Note that,

$$L\{P'_{q,r}(t)\} = -\xi_{q,r} + \theta \cdot P^*_{q,r}(\theta),$$

where

$$\begin{aligned} \xi_{q,r} &= 1, && \text{if } q = r = 1 \\ &= 0, && \text{otherwise.} \end{aligned}$$

Taking transforms on both sides of (17) and (18), and simplifying, we obtain

$$(\theta + a_q + b_r) P^*_{q,r}(\theta) = a_{q-1}\cdot P^*_{q-1,r}(\theta) + b_{r-1}\cdot P^*_{q,r-1}(\theta), \qquad n+1 \geq q > r \geq 1 \tag{20}$$

$$(\theta + a_1) \cdot P^*_{1,1}(\theta) = 1 \tag{21}$$

$$(\theta + a_q) \cdot P^*_{q,q}(\theta) = b_{q-1} \cdot P^*_{q,q-1}(\theta), \qquad n+1 \geq q \geq 2 \tag{22}$$

where $a_{n+1} \equiv b_0 \equiv 0$.

From the definition of the state probabilities given in (14), it follows that $P_{n+1,\,n+1}(t)$ is the distribution function of the total elapsed time with respect to the natural sequence s. (For another application of this concept see Reference [6], pp. 310.) Therefore the corresponding expected elapsed time will be

$$E_s(T) = \int_0^\infty t \cdot \frac{d}{dt} P_{n+1,\,n+1}(t)\, dt.$$

Substituting $P'_{n+1,\,n+1}(t)$ from (18), we get

$$E_s(T) = b_n \cdot \int_0^\infty t \cdot P_{n+1,\,n}\,(t)\,dt$$

or

$$E_s(T) = -\,b_n \cdot \frac{d}{d\theta} P^*_{n+1,\,n}\,(\theta)\,|_{\theta=0}. \tag{23}$$

Expanding $P^*_{q,r}(\theta)$ in ascending powers of θ, we get

$$P^*_{q,r}(\theta) = \alpha_{q,r} + \beta_{q,r} \cdot \theta + 0(\theta^2), \qquad n+1 \geq q \geq r \geq 1, \tag{24}$$

where $\alpha_{q,r}$ and $\beta_{q,r}$ are constants.

It then follows from Equation (23) that

$$E_s(T) = -\,b_n \cdot \beta_{n+1,\,n}. \tag{25}$$

In the spirit of Prabhu [9, pp. 134] we can evaluate the constants $\alpha_{q,r}$ and $\beta_{q,r}$ recursively, for $q=1, 2, \ldots, n+1$ and $r=1, 2, \ldots, n$. Combining (24) with (20) through (22) we get the following sets of recursive relations:

$$\alpha_{1,1} = 1/a_1; \qquad \beta_{1,1} = -\,1/a_1^2; \tag{26}$$

$$\left.\begin{aligned} \alpha_{q,r} &= [1/(a_q + b_r)] \cdot [a_{q-r} \cdot \alpha_{q-1,r} + b_{r-1} \cdot \alpha_{q,r-1}] \\ \beta_{q,r} &= -\,[1/(a_q + b_r)] \cdot [\alpha_{q,r} - a_{q-1}\beta_{q-1,r} - b_{r-1}\beta_{q,r-1}] \end{aligned}\right\} n+1 \geq q > r \geq 1; \tag{27}$$

$$\left.\begin{aligned} \alpha_{q,q} &= [b_{q-1}/a_q] \cdot \alpha_{q,q-1} \\ \beta_{q,q} &= -\,[1/a_q][\alpha_{q,q} - b_{q-1} \cdot \beta_{q,q-1}] \end{aligned}\right\} n \geq q \geq 2; \tag{28}$$

with $a_{n+1} \equiv b_0 \equiv 0$.

With the help of Equations (26), (27), and (28) we can now evaluate the values of $\alpha_{q,r}$ and $\beta_{q,r}$ for all possible values of q and r. Then we can find $\beta_{n+1,\,n}$ and put it into Equation (25) to get $E_s(T)$ for the sequence s. These expressions are somewhat cumbersome and hence are omitted in this paper.

Note that the expected elapsed time for any other sequence s' can be directly obtained from the expression of $E_s(T)$ simply by interchanging the appropriate subscripts.

Computational Aspects

The expected elapsed time $E_s(t)$ for any given sequence s can be easily computed by directly using the recursive relations (27) and (28), instead of handling the cumbersome analytical expression of $E_s(T)$. Thus, for an n-job problem one needs to evaluate only the $n(n+1)$ constants $\alpha_{q,r}$, $\beta_{q,r}$, $n+1 \geq q \geq r \geq 1$, $r \leq n$; and this may be very quickly done on a computer. For example, the evaluation of the expected elapsed time for a 50-job problem takes less than half a second on an IBM MODEL 370/165. The memory requirements are also very small since it is sufficient to store at most $2n$ constants at any time.

THE OPTIMAL VS A HEURISTIC ORDERING RULE

A heuristic approach towards solving the n-job, 2-machine scheduling problem with stochastic process times is to consider the deterministic-equivalent of the problem. That is, solve the corresponding deterministic problem with job times equal to the mean process times. The optimal sequence to this problem is given by Johnson's rule [7, pp. 63]. It is of interest to investigate how close is this heuristic ordering to the optimal sequence.

Makino [8] has made such a study for the specific case of a two-job (two-machine) problem using examples that have a particular structure (viz. $a_1/b_1 = 2$). In this section we will consider n-job problems, and allow the process time parameters to take any feasible values.

The optimal sequence rule (13) maintains that job j precedes job $(j+1)$ if $(a_j + b_{j+1}) \geq a_{j+1} + b_j$

$$\text{i.e., } b'_{j+1} \geq \{b'_j - (1 - a'_{j+1})\}, \tag{29}$$

where

$$b'_{j+1} = b_{j+1}/a_j; \qquad a'_{j+1} = a_{j+1}/a_j; \qquad b'_j = b_j/a_j. \tag{30}$$

Figure 2 depicts the graph defined by (29). Job j should precede job $(j+1)$ for all points above the line, and the ordering is reverse otherwise. For a point lying on the line either sequence is optimal.

According to the heuristic rule job j precedes job $j+1$ if

$$\text{Min}\{(1/a_j), (1/b_{j+1})\} \leq \text{Min}\{(1/a_{j+1}), (1/b_j)\},$$

which, with (30), reduces to

$$\text{Min}\{(1), (1/b'_{j+1})\} \leq \text{Min}\{(1/a'_{j+1}), (1/b'_j)\}. \tag{31}$$

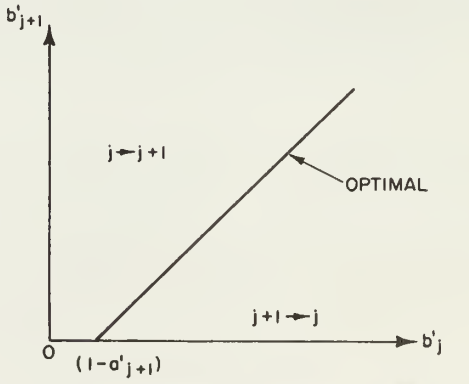


FIGURE 2

The graph of b'_{j+1} with respect to b'_j given by (31) varies in an interesting manner with the value of a'_{j+1}. For $a'_{j+1} > 1$, the shape changes in a regular pattern with a'_{j+1}, but for $a'_{j+1} > 1$, the shape is indifferent to the value of a'_{j+1}. At the critical value $a'_{j+1} = 1$, the graph takes a shape that combines those of the two other cases at their limits. These are displayed in Figures 3, 4, and 5, respectively. On each of these graphs the line corresponding to the optimal ordering has been superimposed. A

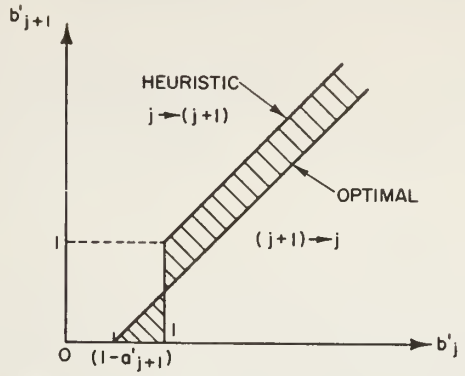


FIGURE 3. $(a'_{j+1} < 1)$

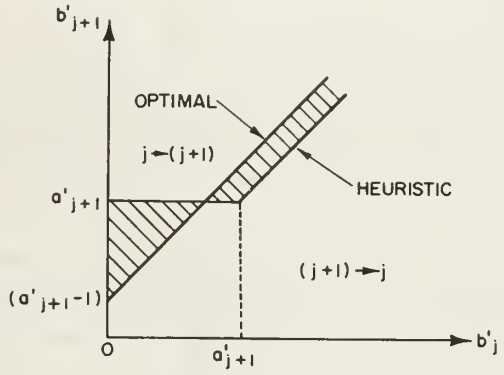


FIGURE 4. $(a'_{j+1} > 1)$

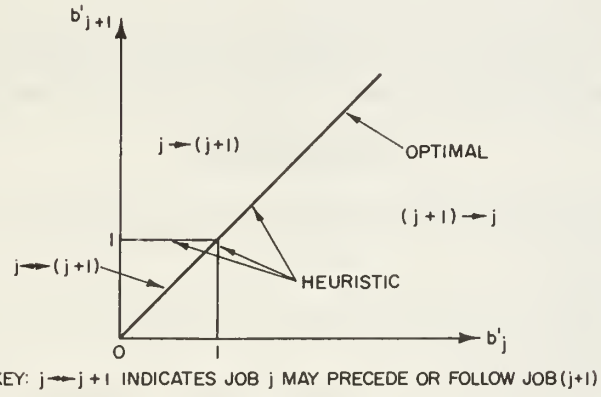


FIGURE 5. $(a'_{j+1} = 1)$

point in the shaded regions of the Figures corresponds to a combination of the process time parameters for which the two methods yield different sequences. Table 1 summarizes the conditions on the four parameters under which the corresponding point will fall within the shaded region. Note that a given set of values of these parameters may correspond to at most one of the four cases.

TABLE 1. *Comparison of the heuristic rule with the optimal solution*

Case No.	a_{j+1}	b_j	b_{j+1}	$a_{j+1}+b_j$	Refer
1a	$< a_j$	$< a_j$		$> (a_j+b_{j+1})$	Figure 3
1b		$> a_j$	$> b_j$	$< (a_j+b_{j+1})$	
2a	$> a_j$		$< a_{j+1}$	$< (a_j+b_{j+1})$	Figure 4
2b		$< b_{j+1}$	$> a_{j+1}$	$> (a_j+b_{j+1})$	

This analysis may now be applied to any pair of the n jobs to ascertain whether their relative precedence is the same under both methods. Thus we may determine how different is the heuristic ordering from that due to the optimal rule.

CONCLUDING COMMENTS

The problem of scheduling n jobs on two machines of a flow shop, where the processing times are exponentially distributed random variables, has been solved. It has also been shown how to evaluate the expected elapsed time for any given sequence. Finding the optimal schedule as well as evaluation of the expected elapsed time for a given schedule poses no computational difficulty for any problem of practical size.

If it is required to calculate the variance of the elapsed time, this can be done using a set of easily formulated recursive relations, similar to those presented earlier for the expected elapsed time.

Considerations of Erlang-K and other non-exponential distributions should lead to useful extensions. Supposition of limited buffer space in between the machines may also give rise to interesting results. It might also be worthwhile to incorporate other objective functions. Such various extensions are currently under investigation.

ACKNOWLEDGMENTS

The authors wish to thank Dr. M. J. M. Posner for his many helpful suggestions. The financial assistance provided through the National Research Council of Canada is also gratefully acknowledged.

REFERENCES

[1] Bagga, P. C., "Sequencing With Random Service Times," Technometrics *12*, 327–334 (May 1970).

[2] Bagga, P. C., "*n*-Job, 2-Machine Sequencing Problems With Stochastic Service Times," Opsearch *7*, 184–197 (Sept. 1970).

[3] Banerjee, B. P., "Single Facility Sequencing With Random Execution Times," Operations Research *13*, 358–364 (May 1965).

[4] Conway, R. C., W. L. Maxwell, and L. W. Miller, *Theory of Scheduling* (Addison Wesley, 1967).

[5] Elmaghraby, S. E., "The Machine Sequencing Problem—Review and Extension," Nav. Res. Log. Quart. *15*, 205–232 (June 1968).

[6] Jaiswal, N. K., "Distribution of Busy Periods for the Bulk-Service Queuing Problem," Defence Science Journal *12*, 309–316 (Oct. 1962).

[7] Johnson, S. W., "Optimal Two- and Three-Stage Production Schedules With Setup Times Included," Nav. Res. Log. Quart. *1*, 61–68 (Mar. 1954).

[8] Makino, T., "On a Scheduling Problem," J. Operations Res. Soc. Japan *8*, 32–44 (Sept. 1965).

[9] Prabhu, N. U., *Stochastic Processes* (MacMillan, 1965).

[10] Talwar, P. P., "A Note on Sequencing Problems with Uncertain Job Times," J. Operations Res. Soc. Japan *9*, 93–97 (July 1967).

PARAMETRIC ANALYSIS OF LINEAR PROGRAMS WITH UPPER BOUNDED VARIABLES

S. S. Panwalkar

Texas Tech University

ABSTRACT

Linear programming problems with upper bounded variables can be solved by regular simplex method by considering upper bounding constraints as explicit constraints of the problem. However, more efficient methods exist which consider these upper bound constraints implicitly. When parametric analysis for problems with upper bounds is to be carried out, one can use the regular parameter analysis by considering the upper bound constraints explicitly. This paper develops formulas for parametric analysis where upper bound constraints are used implicitly, thus reducing the size of the basic matrix.

Consider a linear programming problem with upper bounds defined by:

$$\text{Maximize } x_0 = CX$$

subject to

$$(A, I)X = P_0 \tag{1}$$

$$0 \leqslant X \leqslant U.$$

For the variables with no upper bounds, the corresponding elements of U are assumed to be infinite. This problem has been studied by Dantzig [2] and by Charnes and Lemke [1], and although such a problem can be solved using the simplex method, it has been found to be advantageous to use a special method [2] which utilizes a smaller basis to solve the problem.

Now consider a parametric linear programming problem given by:

$$\text{Maximize } x_0 = (C + \theta E)X$$

subject to

$$(A, I)X = P_0 + \theta R_0 \tag{2}$$

$$X \geqslant 0,$$

where E and R_0 are given constant vectors and θ is a real parameter. The variation with θ of the optimum solution to this problem has been studied by Gass and Saaty [4, 5] and by Manne [7]. The study

is based on the fact that (a) changes in the objective function can only affect the optimality of the problem (or the feasibility of its dual), and (b) changes in the right-hand side of the constraints can only affect the feasibility of the problem (or the optimality of its dual). Thus, by checking these conditions systematically, the characteristic value of θ at which the optimum basic solution changes can be determined. This analysis can also be extended to the parametric variations in the elements of the matrix A in (1) above. The analysis in this case, however, is not as general as in the case of objective function coefficients and the right-hand side elements of the constraints (see for example, Kim [6]).

The same techniques can be extended to a parametric *LP* problem with upper bounds. However, no method exists to take advantage of the structure of upper bounded problem using a smaller basis for the parametric analysis. The following development gives a method to solve the parametric *LP* problem with upper bounds. Thus, in addition to the regular parametric problem, the analysis will include the parameterization of the upper bounds. This parametric variation can be represented by:

$$0 \leqslant X \leqslant U + \theta V, \tag{3}$$

where V is a known constant vector and θ is the common parameter.

VARIATIONS IN P_0 AND U VECTORS

Consider the problem represented by the combination of Equations (2) and (3) with $E=0$. Assume for the moment that $\theta \geqslant 0$. It is also assumed that an optimal solution exists at $\theta=0$.

Solving the problem for $\theta=0$ using the procedure of the bounded-variable problem yields an optimum solution in which $n-m$ variables are set equal to their upper or lower bounds and the remaining m variables have an associated basis B. In the regular simplex method the m basic variables are nonnegative and all other $(n-m)$ variables, called nonbasic, are at zero level. In the bounded variable problem the same concept is required and in order to achieve this, a variable x_i reaching its upper bound u_i is replaced by

$$x_i' = u_i - x_i. \tag{4}$$

Thus, only the m variables associated with the basis are nonnegative and the remaining $n-m$ nonbasic variables will always be at zero level due to the above substitution.

Let U_u be the vector of upper bounds corresponding to the variables X_u which are at the upper bounds (and have been replaced by X_u') in the optimum solution. Let D_u be the corresponding columns from (A, I). Then the values of m basic variables are given by the vector X_B, where

$$0 \leqslant X_B = B^{-1}(P_0 - D_u U_u) \leqslant U_B, \tag{5}$$

and U_B is the vector of upper bounds for the corresponding elements of X_B.

For $\theta > 0$, the solution will remain optimum as long as

$$0 \leqslant X_B = B^{-1}[P_0 - D_u(U_u + \theta V_u) + \theta R_0] \leqslant U_B + \theta V_B, \tag{6}$$

where V_u and V_B are defined in a similar manner. Let

$$P = P_0 - D_u U_u \text{ and } R = R_0 - D_u V_u.$$

Then the Inequality (6) can be written as

(7) $$0 \leq X_B = B^{-1}(P + \theta R) \leq U_B + \theta V_B.$$

Let $(B^{-1}P)_i$ and $(B^{-1}R)_i$ denote the ith elements of the respective column vectors. Also, let U_{B_i} and V_{B_i} denote the values of U and V corresponding to the basic variable in the ith row. Then,

(i) If ϕ_1' represents the minimum value of θ at which an upper bound of a basic variable will be exceeded, this will be given by

(8) $$\phi_1' = \min_i \left\{ \frac{U_{B_i} - (B^{-1}P)_i}{(B^{-1}R)_i - V_{B_i}} \right\} \qquad \text{for } (B^{-1}R)_i - V_{B_i} > 0.$$

If all $(B^{-1}R)_i - V_{B_i} \leq 0$, then let $\phi_1' = \infty$. Also, if corresponding to every $(B^{-1}R)_i - V_{B_i} > 0$, the basic variable X_{B_i} does not have a finite upper bound $(U_{B_i} = \infty)$, then ϕ_1' will be infinite.

(ii) If $(B^{-1}R)_i < 0$ for at least one value of i, then the corresponding nonnegativity constraint will be violated at some value of θ. Let

(9) $$\phi_1'' = \min_i \left\{ \frac{-(B^{-1}P)_i}{(B^{-1}R)_i} \right\} \qquad \text{for } (B^{-1}R)_i < 0.$$

If all $(B^{-1}R)_i \geq 0$, then let $\phi_1'' = \infty$.

Let

(10) $$\alpha = \min_j \left\{ \frac{-u_j}{v_j} \right\} \qquad \text{for } v_j < 0$$
$$= \infty \qquad \text{if all } v_j \geq 0.$$

Let $\theta_0 = 0$ and let θ_1 denote the first characteristic value of θ (meaning the value of θ at which the basis for optimal solution changes). If $\phi_1 = \min(\phi_1', \phi_1'')$ then θ_1 will be given by

(11) $$\theta_1 = \theta_0 + \phi_1,$$

provided $\theta_1 < \alpha$. If $\theta_1 \geq \alpha$ the problem becomes inconsistent at $\theta = \alpha$ and no characteristic value exists.

For the problem without upper bounds, Gass and Saaty [4, 5] show that the optimal basis at $\theta = 0$ represents the optimal basis for any θ in the range $0 \leq \theta \leq \theta_1$. At $\theta = \theta_1$ a new basis must be obtained by carrying out a simplex iteration to obtain the new optimum solution. For the bounded variable problem this is done as follows.

If $\phi_1 = \infty$, there is no characteristic value for the problem. The basis at $\theta = 0$ will be an optimum basis for all values of $\theta \leq \alpha$ and the analysis is stopped. If ϕ_1 is finite and assuming $\phi_1 = \theta_1 \leq \alpha$, calculate

$$X_B = B^{-1}[P_0 - D_u(U_u + \phi_1 V_u) + \phi_1 R_0]. \tag{12}$$

At least one element of X_B must be zero at $\theta = \phi_1$. Substitute the value of the objective function by

$$x_0 = C_B X_B + C_u(U_u + \phi_1 V_u),$$

where C_B represent the coefficients in the objective function corresponding to the basic variables and C_u corresponding to the variables at upper bound. A dual simplex iteration is carried out and the new basis and new solution is obtained. If the dual iteration cannot be carried out, the solution space does not have a feasible point for $\theta > \theta_1$ and the analysis is terminated. If, however, the dual iteration is carried out the analysis is continued. For this the following must be defined.

Let, B^1 be the new basis after the 1st characteristics value θ_1.

$$P_0^1 = P_0 + \theta_1 R_0$$

$$U^1 = U + \theta_1 V$$

$$P^1 = P_0^1 - D_u^1 U_u^1$$

and

$$R^1 = R_0 - D_u^1 V_u^1.$$

In the above, D_u^1 represents the matrix of column vectors from (A, I) which are at their upper bounds after the new basis has been calculated and V_u^1, the corresponding V values.

In general let θ^k represent the kth characteristic value (it has been shown in [4, 5] that there are a finite number of characteristic values).

Then

$$\theta_{k+1} = \theta_k + \phi_{k+1}, \tag{13}$$

where

$$\phi_{k+1} = \min\,(\phi'_{k+1}, \phi''_{k+1}),$$

and

$$\phi'_{k+1} = \min_i \left\{ \frac{U_B^k - [(B^k)^{-1}P^k]_i}{[(B^k)^{-1}R^k]_i - V_{B_i^k}} \right\} \quad \text{for the denominator} > 0, \tag{14}$$

and

$$\phi''_{k+1} = \min_i \left\{ \frac{-[(B^k)^{-1}P^k]_i}{[(B^k)^{-1}R^k]_i} \right\} \quad \text{for the denominator} < 0. \tag{15}$$

If, however, $\theta_{k+1} > \alpha$, there is no $k+i$th characteristic value and the analysis is terminated. In (14) and (15) the notation $[\ \ ]_i$ denotes the ith term of the column vector $[\ \ ]$. It may be noted that for $k = 0$,

Equations (14) and (15) reduce to Equations (8) and (9), respectively. However, the superscript zero was omitted for simplicity. At $\theta=\theta_{k+1}$ the value of X_{Bk} is given by

$$X_{Bk}=(B^k)^{-1}[P^k+\phi_{k+1}R^k], \tag{16}$$

and the new basis X_{Bk+1} is obtained through a dual iteration.

EXAMPLE 1: Solve the following parametric problem $\theta \geq 0$:

$$\text{Maximize } x_0=3x_1+5x_2+2x_3,$$

subject to

$$x_1+x_2+2x_3+x_4=10-\theta$$

$$2x_1+4x_2+3x_3+x_5=16+2\theta$$

$$0 \leq x_1 \leq 4+\theta, \qquad 0 \leq x_2 \leq 3+\theta, \qquad 0 \leq x_3 \leq 3-(1/4)\theta.$$

SOLUTION: Note that

$$R_0=\begin{pmatrix}-1\\2\end{pmatrix}$$

and

$$U+V=\begin{pmatrix}4\\3\\3\\\infty\\\infty\end{pmatrix}+\theta\begin{pmatrix}1\\1\\-1/4\\0\\0\end{pmatrix}.$$

At $\theta=0$ the optimum solution can be easily obtained and is given by

	x_1'	x_5	x_3	Solution
	1/2	5/4	7/4	22
x_4	−1/2	−1/4	5/4	4
x_2	−1/2	1/4	3/4	2

The basic variables in the tableau are $x_4=4$ and $x_2=2$. All the nonbasic variables x_1', x_5, and x_3 have zero values. The value of x_1 can be easily obtained by

$$x_1=u_1-x_1'=4-0=4.$$

Thus x_1 is at its upper bound. The value of the objective function is $x_0 = 22$. Also B^{-1} is given by the columns for variables x_4 and x_5. Thus

$$B^{-1} = \begin{pmatrix} 1 & -1/4 \\ 0 & 1/4 \end{pmatrix}$$

$$D_u U_u = \begin{pmatrix} 1 \\ 2 \end{pmatrix} (4) = \begin{pmatrix} 4 \\ 8 \end{pmatrix},$$

$$P = P_0 - D_u U_u = \begin{pmatrix} 10 \\ 16 \end{pmatrix} - \begin{pmatrix} 4 \\ 8 \end{pmatrix} = \begin{pmatrix} 6 \\ 8 \end{pmatrix}$$

Hence

$$B^{-1}P = \begin{pmatrix} 4 \\ 2 \end{pmatrix},$$

$$R = R_0 - D_u V_u = \begin{pmatrix} -1 \\ 2 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \end{pmatrix} (1) = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$$

Thus

$$B^{-1}R = \begin{pmatrix} -2 \\ 0 \end{pmatrix}.$$

From these,

$$B^{-1}R - V_B = \begin{pmatrix} -2 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ -1 \end{pmatrix} < 0,$$

which gives

$$\phi_1' = \infty.$$

Also

$$\phi_1'' = \min \left\{ \frac{-4}{-2}, \text{——} \right\} = 2.$$

Hence

$$\phi_1 = \min (\infty, 2) = 2$$

and

$$\theta_1 = 0 + 2 = 2$$

$$\alpha = \frac{-3}{-1/4} = 12 > \theta_1.$$

Hence the first characteristic value exists. At $\theta = 2$, the values of X_B are given by (16).

$$X_B = \begin{pmatrix} 1 & -1/4 \\ 0 & 1/4 \end{pmatrix} \left[\begin{pmatrix} 10 \\ 16 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \end{pmatrix} (4 + 2.1) + 2 \begin{pmatrix} -1 \\ 2 \end{pmatrix} \right]$$

$$= \begin{pmatrix} 1 & -1/4 \\ 0 & 1/4 \end{pmatrix} \begin{pmatrix} 2 \\ 8 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

The value of the objective function at $\theta = 2$ can be easily calculated and is 28. Thus the tableau at $\theta = 2$ with the old basis will be

	x_1'	x_5	x_3	
	1/2	5/4	7/4	28
x_4	−1/2	−1/4	5/4	0
x_2	−1/2	1/4	3/4	2

Note that $x_1 = (u_1 + \theta v_1) - x_1' = 6 - 0 = 6$. Carrying out the dual iteration, the new solution with new basis is given below.

	x_4	x_5	x_3	
	1	1	3	28
x_1'	−2	1/2	−5/2	0
x_2	−1	1/2	−1/2	2

Now,

$$(B^1)^{-1} = \begin{pmatrix} -2 & 1/2 \\ -1 & 1/2 \end{pmatrix}$$

$$P_0^1 = P_0 + \theta_1 R_0 = \begin{pmatrix} 10 \\ 16 \end{pmatrix} + 2\begin{pmatrix} -1 \\ 2 \end{pmatrix} = \begin{pmatrix} 8 \\ 20 \end{pmatrix}$$

$$U_u^1 = 4 + 2(1) = 6$$

$$P^1 = \begin{pmatrix} 8 \\ 20 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \end{pmatrix}(6) = \begin{pmatrix} 2 \\ 8 \end{pmatrix}$$

$$R^1 = \begin{pmatrix} -1 \\ 2 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \end{pmatrix}\begin{pmatrix} 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$$

$U_{B^1}^1 =$ (New upper bounds for x_1' and x_2)

$$= \begin{pmatrix} 6 \\ 5 \end{pmatrix}$$

$$V_{B^1} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$(B^1)^{-1}P^1 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \quad (B^1)^{-1}R^1 = \begin{pmatrix} 4 \\ 2 \end{pmatrix} > 0$$

$$(B^1)^{-1}R^1 - V_{B^1} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

Thus

$$\phi_2' = \min\left\{\frac{6-0}{3}, \frac{5-2}{1}\right\} = 2$$

$$\phi_2'' = \infty,$$

and

$$\phi_2 = 2,\ \theta_2 = 2+2 = 4 < \alpha = 12$$

$$X_{B^2} = \begin{pmatrix} -2 & 1/2 \\ -1 & 1/2 \end{pmatrix}\left[\begin{pmatrix} 2 \\ 8 \end{pmatrix} + 2\begin{pmatrix} -2 \\ 0 \end{pmatrix}\right]$$

$$= \begin{pmatrix} -2 & 1/2 \\ -1 & 1/2 \end{pmatrix}\begin{pmatrix} -2 \\ 8 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \end{pmatrix}.$$

At this point the alternative tableau can be calculated and the analysis can be carried out beyond $\theta = 4$. Note that the analysis in any case will be stopped beyond $\theta = 10$, because the first constraint cannot be satisfied by any $x_i > 0$. This will be indicated in the solution because at $\theta = 10$, a dual iteration cannot be carried out.

VARIATIONS IN P_0 AND U VECTORS FOR $\theta < 0$

From (5) and (7) it may be noted that

$$0 \leq X_B = B^{-1}P \leq U_B$$

and

$$0 \leq X_B = B^{-1}P + \theta B^{-1}R \leq U_B + \theta V_B.$$

Since θ is now negative, the upper bound of X_{B_i} will be exceeded if $(B^{-1}R)_i - V_{B_i} < 0$. Also if $(B^{-1}R)_i > 0$ for at least one i, the corresponding nonnegativity constraint will be violated. Thus, letting $\theta_0' = 0$ and denoting kth characteristic value of θ by θ_k'

(17) $$\theta_{k+1}' = \theta_k' + \lambda_{k+1},$$

where

(18) $$\lambda_{k+1} = \max\ (\lambda_{k+1}', \lambda_{k+1}''),$$

with

(19) $$\lambda_{k+1}' = \max_i \left\{\frac{U_{B_i}^k - [(B^k)^{-1}P^k]_i}{[(B^k)^{-1}R^k]_i - V_{B_i^k}}\right\} \quad \text{for the denominator} < 0,$$

and

$$\lambda''_{k+1} = \max_i \left\{ \frac{-[(B^k)^{-1}P^k]_i}{[(B^k)^{-1}R^k]_i} \right\} \quad \text{for the denominator} > 0. \tag{20}$$

Also, $\theta'_{k+1} \geq \alpha$ in order that the problem has consistant upper bound constraints. The value of α is given by

$$\alpha = \max \left\{ \frac{-u_j}{v_j} \right\} \quad \text{for } v_j > 0. \tag{21}$$

VARIATIONS IN C AND U VECTORS

For this discussion assume $\theta > 0$. The problem will be represented by (2) and (3) with $R_0 = 0$. It has been already stated that changes in C will affect only optimality of the primal. Similarly, changes in U will only affect the feasibility of the primal. With this information the following procedure illustrates the calculation for θ_1 only. The general calculations for θ_k and for $\theta < 0$ require simple extensions.

Let there be an optimum solution at $\theta = 0$. This means that all $z_j - c_j \geq 0$, i.e.,

$$C_B B^{-1} P_j - c_j \geq 0, \tag{22}$$

where P_j denotes the column of jth variable from (A, I) matrix. If the same basis is optimum for $\theta > 0$,

$$(C_B + \theta E_B)\, B^{-1} P_j - (c_j + \theta e_j) \geq \mathbf{0}$$

or

$$\theta(E_B B^{-1} P_j - e_j) \geq (C_B B^{-1} P_j - c_j). \tag{23}$$

Thus if $(E_B B^{-1} P_j - e_j) \geq 0$ for all j, the solution will be optimum for all values of θ.

If $(E_B B^{-1} P_j - e_j) < 0$ for at least one j, then denote

$$\phi'''_1 = \min_j \left\{ \frac{-(C_B B^{-1} P_j - c_j)}{(E_B B^{-1} P_j - e_j)} \right\} \quad \text{for the denominator} < 0. \tag{24}$$

Also for feasibility, at $\theta = 0$, from (5)

$$0 \leq X_B = B^{-1} P \leq U_B,$$

and for $\theta > 0$, from (6)

$$0 \leq X_B = B^{-1}(P - \theta D_u V_u) \leq U_B + \theta V_B.$$

This leads to two values ϕ'_1 and ϕ''_1. Then

$$\phi_1 = \min\,(\phi'_1, \phi''_1, \phi'''_1) \tag{25}$$

and

$$\theta_1 = \theta_0 + \phi_1 \qquad \text{with } \theta_0 = 0.$$

If $\phi =$ either ϕ_1' or ϕ_1'' carry out a dual iteration. If $\phi = \phi_1'''$, optimality is violated; carry out a primal iteration using bounded variable technique. Also, the value of α calculated from (10) must be greater than θ_1.

SIMULTANEOUS VARIATIONS IN C, P_0, AND U

In this case ϕ_1''' is calculated from (24), ϕ_1' and ϕ_2'' from (8) and (9), and α from (10). This is possible because variations in C do not affect feasibility and the variations in P_0 and U do not affect optimality. The formulas are not quoted here because they can be directly obtained from the previous sections.

GENERAL COMMENTS

When there are variations in P_0 and/or U only, there will be a need for carrying out a dual simplex iteration. Since the pivoting is done in the row with zero right hand element, there is no need to check upper bound violation. A regular dual iteration is carried out. For the variations in C, a primal iteration is required. This is similar to the one in bounded variable technique [2]. This means that the available computer programs can be used to carry out these iterations. Also, since the bounded variable techniques are more efficient than the regular simplex method (for bounded variable problems), it is expected that the use of the formulas presented in this paper in conjunction with bounded variable techniques would result in efficient parametric analysis.

ACKNOWLEDGMENTS

The author wishes to express his gratitude to the referee for his insight and helpful comments about this paper.

BIBLIOGRAPHY

[1] Charnes, A. and C. E. Lemke, "The Bounded Variables Problem," ONR Reference Memorandum 10, Graduate School of Industrial Administration, Carnegie Institute of Technology, Pittsburgh, Pa. (1954).

[2] Dantzig, G. B., "Upper Bounds, Secondary Constraints and Triangularity in Linear Programming," Econometrica, Vol. 23, No. 2 (1955).

[3] Dantzig, G. B., *Linear Programming and Extensions* (Princeton University Press, Princeton, New Jersey, 1963).

[4] Gass, S. I. and T. L. Saaty, "The Parametric Objective Function, Part I," Operations Research, Vol. 2 (1954).

[5] Gass, S. I. and T. L. Saaty, "The Parametric Objective Function, Part II: Generalization," Operations Research, Vol. 3 (1955).

[6] Kim, C., "Parameterizing an Activity Vector in Linear Programming," Operating Research, Vol. 19, No. 7 (1971).

[7] Manne, A. S., *Scheduling of Petroleum Refinery Operations* (Harvard University Press, Cambridge, Massachusetts, 1956).

[8] Panwalkar, S. S., "A Primal-Dual Algorithm for Bounded Variables with Extensions to Parametric Analysis," unpublished Doctoral Dissertation, University of Oklahoma (1970).

[9] Taha, H. A., *Operations Research: An Introduction* (Macmillan Company, New York, 1971).

[10] Wagner, H. M., "The Dual Simplex Algorithm for Bounded Variables," Nav. Res. Log. Quart. 5, 257–261 (1958).

A COMPARISON OF THE PRIMAL-SIMPLEX AND COMPLEMENTARY PIVOT METHODS FOR LINEAR PROGRAMMING

Arunachalam Ravindran

School of Industrial Engineering
Purdue University

ABSTRACT

A comparison of the complementary pivot method of Lemke-Howson and the more commonly used primal-simplex method for solving linear programming problems in symmetric dual form has been made. In our tests the complementary pivot method shows a definite superiority over the simplex method both with regard to the number of iterations and computation time.

The simplex method for linear programming developed by G. B. Dantzig [2] is now extensively used to solve linear programs. A possible inefficiency in this method is the need for an initial basic feasible solution to start with. If none is readily available one uses the method of artificial variables which leads to solving, in fact, another linear program.

Another method came into existence when Cottle and Dantzig [1] proposed a complementary problem of the form:

Find $w \geqq 0, z \geqq 0$

(1) such that $w = Mz + q, w'z = 0,$

where M is a square matrix and w, z, q are vectors. They showed that a linear program can be transformed to the above form and a solution to (1) gives an optimal solution to the linear program. Lemke and Howson [4, 5] proposed a complementary pivot method to solve system (1). Unlike the simplex method, finding a starting solution is much easier for the complementary pivot method. Since this method has not been extensively explored or applied, a study has been made to compare the relative efficiencies of these two methods to solve linear programs. A brief discussion of the complementary pivot method of Lemke-Howson is given in section 1. A detailed discussion is available in Reference [7].

1. THEORETICAL BACKGROUND

Consider a linear program in symmetric dual form:

$$\text{Minimize } c'x$$

$$\text{subject to } Ax \geqq b, \qquad x \geqq 0,$$

where A is an $(m \times n)$ matrix, c and x are $(n \times 1)$ column vectors and b is an $(m \times 1)$ column vector. ('Prime' is used to denote the transpose of a vector or matrix.)

An optimum solution to the above problem may be obtained by solving a Kuhn-Tucker problem of the form:

$$\begin{bmatrix} v \\ u \end{bmatrix} = \begin{bmatrix} 0 & -A' \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ -b \end{bmatrix}$$

$$u, v, x, y \geqq 0$$

$$v'x + u'y = 0, \tag{2}$$

where u denotes the slack variables of the given linear program and (y, v) denotes the variables of the dual problem. In other words, system (2) represents primal-dual feasibility and the complementary slackness conditions. Comparing system (2) with the original complementary problem (1), we have:

$$w = \begin{pmatrix} v \\ u \end{pmatrix}, z = \begin{pmatrix} x \\ y \end{pmatrix}, M = \begin{bmatrix} 0 & -A' \\ A & 0 \end{bmatrix} \text{ and } q = \begin{pmatrix} c \\ -b \end{pmatrix}.$$

It should be remarked here that for a linear program M will be a square, nonsymmetric, positive, semidefinite matrix of order $(m + n)$. Hence, to solve system (2), the complementary pivot method of Lemke [4] can be used which guarantees to terminate with a "complementary solution" when M is a positive semidefinite matrix (a complementary solution to system (1), is defined as a solution (w, z) satisfying $(w, z) \geqq 0$, $w = Mz + q$, and $w'z = 0$).

If the q vector is nonnegative, then we have a trivial complementary solution given by $w = q$, $z = 0$. Hence, Lemke-Howson's method starts with the assumption that there exists at least one $q_i < 0$ and the initial basic solution $w = q$, $z = 0$ is complementary, but is infeasible. At the first iteration, an artificial variable z_0 is added to each equation at a sufficiently positive level such that $w = q + e'z_0 \geqq 0$ where $e = (1, 1, \ldots, 1)$. Then, at each iteration, the method employs a 'simplex type' ratio rule to go to a new basic solution maintaining the complementary property that $w'z = 0$ but decreasing the value of z_0. The process terminates when z_0 becomes zero indicating a complementary solution or with an indication that there is no solution to the given complementary problem.

2. RANDOM GENERATION OF DATA

The two methods are compared by testing their efficiencies with regard to the number of iterations and computation time on some sample linear programming problems. A number of linear programs were randomly generated of the form: $Ax \geqq b$, $x \geqq 0$ and minimize $c'x$. A computer code was written to generate the coefficients of A, b, and c for different sets of problems varying in size (rows and columns) using a random number generator. The number of constraints, the number of variables, the percentage of nonzero elements in the coefficient matrix were the input parameters to the computer code. By use of this information, a random number was chosen from a uniform distribution between 0 and 100 for the nonzero elements. We decided to use this random generation procedure for the test problems so that it will be in line with the two previous experimental studies done in solving linear programs by Kuhn and Quandt [3] and Mueller and Cooper [6].

In order to solve these problems by simplex method, the existing library routines supplied by

IBM and CDC were used. For Lemke's complementary pivot method, a computer program in FORTRAN IV was developed by the author. The original Lemke-Howson's algorithm was modified along the lines of the revised simplex method so that the pivot operations are done only to the inverse of the basis rather than to the entire tableau.

3. RESULTS

More than 170 randomly generated linear programs were run to compare the iteration counts and execution times of both the methods. These problems were divided into 17 different sets varying in size and other characteristics. The results obtained, as well as some calculated results, are tabulated in Tables I and II.

TABLE I. *Results of the Study: Run 1*

Problem set	Matrix A		Mean iteration count		Mean execution time (sec)		Ratio = Simplex / Lemke		Execution time / iteration count	
	Size	Type	Lemke	Simplex	Lemke	Simplex	Iteration	Time	Lemke	Simplex
1	5 × 5	all > 0	7.4	11.3	.....	.....	1.52	.....	.....	.....
2	10 × 10	all > 0	11.8	26.9	1.5	4.8	2.27	3.2	0.127	0.178
3	15 × 15	all > 0	16.6	48.1	2.6	9.0	2.89	3.46	0.157	0.187
4	15 × 15	50% > 0 50% < 0	33	30	3.4	7.2	0.91	2.12	0.103	0.204
5	15 × 15	80% = 0 20% > 0	24	33.2	2.8	7.8	1.38	2.78	0.117	0.235
6	15 × 15	80% = 0 10% > 0; 10% < 0	10.4	15.2	2	5.4	1.46	2.7	0.192	0.355
7	15 × 15	50% > 0 50% < 0	23.5	24	3.1	6.6	1.02	2.13	0.132	0.275
8	15 × 15	80% = 0 20% > 0	10.2	21	2.8	7.9	2.06	2.82	0.284	0.376

NOTES:

1. All calculations were done on IBM 7094 computer at U.C., Berkeley.
2. Ten problems were tested in each problem set.
3. All the problems in Run 1 had positive coefficients for the vectors b and c.
4. Problem sets 6, 7, and 8 contain all infeasible linear programs.

It is evident from an examination of Tables I and II that the complementary pivot method consistently requires fewer iterations and less execution time. For those problem sets where the coefficient matrix contains all positive elements, the complementary pivot method is far superior, requiring as little as one-half to one-third the iterations and execution time of simplex. Since many applications of linear programming contain more zeros in the coefficient matrix, we included a number of problem sets with a higher percentage of zeros. Furthermore, we wanted to analyze the behavioral pattern of both the methods as the number of zeros becomes larger. An examination of the results in Table I and Table II suggest that Lemke's method is more sensitive to the density of the coefficient matrix compared to the simplex method. As a matter of fact, Dantzig [1] has pointed out that the number of iterations of the simplex method is sensitive to the number of rows only. This is confirmed by our results for the mean iteration count in Table II for the simplex method. In spite of the sensitiveness of Lemke's method with regard to the density of the matrix, it is evident from the tables that Lemke's method consistently did better than the simplex method.

TABLE II. *Results of the Study: Run 2*

Problem set	Matrix A		Vector b	Vector C	Mean iteration count		Mean execution time (sec.)		Ratio = Simplex / Lemke		Execution time / Iteration count	
	Size	Type			Lemke	Simplex	Lemke	Simplex	Iteration	Time	Lemke	Simplex
1	10×15	all > 0	all > 0	all > 0	12.4	30.3	0.652	1.95	2.44	2.99	0.0526	0.0645
2	10×15	50% > 0; 25% < 0; 25% = 0	all > 0	all > 0	26.4	24.6	0.935	1.64	0.932	1.75	0.0354	0.0667
3	10×15	33% > 0; 33% < 0; 34% = 0	all > 0	all > 0	22.8	22.5	0.949	1.53	0.986	1.61	0.0416	0.0682
4	10×15	25% > 0; 25% < 0; 50% = 0	all > 0	all > 0	21.7	22.2	0.834	1.45	1.02	1.74	0.039	0.0655
5	10×15	50% > 0 50% < 0	50% > 0 50% < 0	all > 0	21.8	26.4	0.666	1.81	1.21	2.72	0.039	0.069
6	10×15	30% > 0 70% = 0	all > 0	all > 0	14.9	23.3	0.849	1.36	1.57	1.6	0.045	0.059
7	10×10	all > 0	50% > 0 50% < 0	all > 0	9.2	23.1	0.402	1.08	2.51	2.68	0.044	0.047
8	10×10	40% > 0; 40% < 0; 20% = 0	all > 0	50% > 0 50% < 0	9.8	19.4	0.41	0.876	1.98	2.14	0.042	0.045
9	10×10	50% > 0; 20% < 0; 30% = 0	all > 0	all > 0	19.6	21.6	0.529	0.953	1.1	1.8	0.027	0.044

NOTES:

1. All calculations were done on CDC 6500 computer at Purdue.
2. Ten problems were tested in each problem set.

Since the test problems in Table I contained only square matrices, it was decided to test these two methods on some elongated problems where the number of columns is greater than the number of rows (Table II). Even under these conditions, the results indicate a clear superiority of Lemke's method.

When the study was first undertaken, it was felt that a possible inefficiency of the simplex method is the need for an initial basic feasible solution. To test this, a comparison of the phase 1 simplex method with Lemke's approach was needed. Since the Linear Programming computer code did not have any option of getting a count of the number of phase 1 iterations, we decided to get this by including some test problems which were infeasible (sets 6, 7, and 8 in Table I). An examination of the results for these problem sets clearly indicates that the complementary pivot method is able to determine the infeasibility faster than the simplex method. This leads us to conjecture that the inefficiency of the simplex method lies possibly in its inability to find an initial basic feasible solution soon.

The referee has pointed out that besides the mean iteration count and execution time, the variance figures are also important. Hence, the variance of the iteration count* was calculated for each problem set.

In order to get a meaningful comparison, the coefficient of variation was computed as:

$$C.V. = \frac{\sqrt{\text{Variance}}}{\text{Mean}} \times 100\%.$$

For Lemke's method, the coefficient of variation for the various problem sets ranged from 9.6 to 25.8 percent, while for the simplex method this figure was from 7.8 to 19.7 percent. Since these figures are not large, we conclude that the results of the data analysis indicate a definite superiority of the complementary pivot method over primal-simplex.

The last two columns in the tables were computed by taking the simple ratio of the mean execution time over the mean iteration count for each method. It should be pointed out here that this mean time includes other extraneous programming executions and hence it does not truly represent the time per iteration. But it is a rough approximation to the mean iteration time and their values are interesting to study. A comparison of the last two columns in both the tables show that the simplex method takes much longer time for each iteration. One of the reasons is the fact that the "column-selection rule" is complicated for the simplex method as one has to compute the simplex multipliers and then look for a column with negative reduced cost. But, for the complementary pivot method, the absence of an objective function in system (1) gives a simple rule for selecting the column at each iteration.

Of course, the complementary pivot method has its own drawbacks. The main drawback is its use of a bigger size matrix for its coefficient matrix, creating a huge storage problem for large scale linear programs. The author attempted to solve this by modifying the complementary pivot method as explained in Reference [7]. But it turned out that this can be done only at the expense of simple column and row selection rules given by Lemke-Howson. So we conclude that as long as we do not anticipate storage problems, it is never undesirable to use the complementary pivot method instead of primal-simplex.

---

*Since all 10 problems in each set were solved together, no breakdown of time for individual problems was available. Hence variance figures for execution times could not be computed.

4. CONCLUDING REMARKS

In conclusion, we would like to suggest a few experiments that could be conducted to further extend our results. With the encouraging results we have for smaller size problems, it would be worthwhile to compare these two methods for larger size problems, especially elongated problems with much more columns than rows. As the referee has aptly pointed out, the simplex method is more sensitive to the number of rows than to the number of columns while the "size" of the Lemke-Howson version goes up as the sum of rows and columns. Hence a comparison of these two methods for elongated problems would be useful and of practical interest.

Almost all the sample problems chosen for the study did not have an obvious feasible starting point. It would be interesting to compare these two methods on some problems with obvious basic feasible solutions. This could then be used to verify whether the inefficiency of the simplex method lies, in fact, in its inability to find a feasible solution soon.

ACKNOWLEDGMENTS

The author wishes to express his appreciation to David Gale of the University of California, Berkeley for introducing him to this topic. I would also like to thank my students, Y. P. Esrock, T. G. Godfrey, E. J. Kannel, B. R. Marks, D. R. Payne, W. W. Peck, D. L. Shunk, and J. A. Tompkins for their assistance in the second computer experiment performed at Purdue. I am grateful to the Editor of *Opsearch* for allowing me to include Table I from Reference [7]. My acknowledgments are due the referee whose careful critique improved the paper considerably.

REFERENCES

[1] Cottle, R. W. and G. B. Dantzig, "Complementary Pivot Theory of Mathematical Programming," J. Linear Algebra and Appl. *1*, 103–125 (1968).

[2] Dantzig, G. B. *Linear Programming and Extensions* (Princeton University Press, Princeton, N.J., (1963).

[3] Kuhn, H. W. and R. E. Quandt, "An Experimental Study of the Simplex Method," Proc. Sympos. Appl. Math. *15*, 107–124 (1963).

[4] Lemke, C. E., "Bi-Matrix Equilibrium Points and Mathematical Programming," Man. Sci *11*, 681–689 (1965).

[5] Lemke, C. E. and J. T. Howson, "Equilibrium Points of Bi-Matrix Games," J. Soc. Indust. Appl. Math. *12*, 413–423 (1964).

[6] Mueller, R. K. and L. Cooper, "A Comparison of Primal Simplex and Primal-Dual Algorithm for Linear Programming," Comm. of ACM *8*, 682–686 (1965).

[7] Ravindran, A. "Computational Aspects of Lemke's Complementary Algorithm Applied to Linear Programs," Opsearch 7, 241–262 (1970).

APPLICATIONS OF MILLS' DIFFERENTIAL

O. Moeschlin and G. Bol

University of Karlsruhe
Federal Republic of Germany

ABSTRACT

Using Mills Differential a game value function Φ, used in the Kemeny-Morgenstern-Thompson growth model to show the existence of solutions, is discussed. In doing so, several proofs become shorter, in addition known results are seen in a new context. Finally connections between the existence of a so-called economic solution and the differentiability or non-differentiability of function Φ are analyzed.

INTRODUCTION

In order to prove the existence of solutions to an economic growth model in [5] a function Φ is discussed. By Φ the game value of the matrix game M_α is assigned to the parameter α:

$$\Phi(\alpha) = \max_{x \in S^m} \min_{y \in S^n} xM_\alpha y = \min_{y \in S^n} \max_{x \in S^m} xM_\alpha y = \Delta(M_\alpha), \tag{1}$$

where $M_\alpha = B - \alpha A$ and $B, A \geqq 0$ are two $(m \times n)$-matrices,

$$S^m = \left\{ x \in R^m \;\middle|\; \sum_{i=1}^{m} x_i = 1 \right\}, \qquad S^n = \left\{ y \in R^n \;\middle|\; \sum_{j=1}^{n} y_j = 1 \right\}.$$

A solution of the model $(\bar{x}, \bar{y}, \bar{\alpha})$, $\bar{x} \in S^m$, $\bar{y} \in S^n$, $\bar{\alpha} \in R^+$ is found if

$$\Phi(\bar{\alpha}) = 0 \tag{2}$$

and

$$\bar{x} B \bar{y} > 0. \tag{3}$$

DEFINITION: If condition (2) is satisfied for a certain α, we are speaking of a feasible growth factor.

DEFINITION: If for a feasible α there exists a pair of strategies (x, y) such that (3) is satisfied, we are speaking of an economic solution.

It has been shown that Φ is continuous and monotone nonincreasing, e.g., in [10].

If two further conditions, introduced by Kemeny, Morgenstern, and Thompson in [5], are provided:

$$\Delta(B) > 0 \qquad [KMT\ 1]$$

$$\Delta(-A) < 0 \qquad [KMT\ 2]$$

(by $\Delta(B)$, $\Delta(-A)$ resp. we mean the game values of B, $(-A)$ resp.), it can be shown that

$$\Phi(0) > 0$$

$$\Phi(\alpha) \to -\infty \qquad \text{for } \alpha \to +\infty.$$

Therefore we conclude:

$$\exists \alpha_1, \alpha_2 \epsilon R^+, \qquad \alpha_2 \leqq \alpha_1$$

$$\Phi(\alpha) > 0 \qquad \text{for } \alpha < \alpha_2$$

$$\Phi(\alpha) = 0 \qquad \text{for } \alpha_2 \leqq \alpha \leqq \alpha_1$$

$$\Phi(\alpha) < 0 \qquad \text{for } \alpha > \alpha_1.$$

DEFINITION: We shall call α_1 the highest, resp. α_2 the lowest feasible growth factor.

Finally Thompson proved in [9]: To α_i, $i \epsilon \{1, 2\}$ there exist vectors $x^{(i)}$, $y^{(i)}$, such that

$$x^{(i)} B y^{(i)} > 0.$$

It is absolutely possible that there may exist further feasible growth vectors associated with economic solutions. Function Φ has been discussed in several papers, e.g., in [5] and [3] in order to prove the existence of solutions to the growth model. In [1] and [4] function Φ is used to calculate α_1 or α_2 and in [10] and [11] it is treated on its own.

In the present paper an analysis of function Φ and therefore an analysis concerning the solutions of the model is given by means of a surrogate of a differential for game values introduced by Mills. It is in the nature of things, that this approach will restate some known results.

APPLICATIONS OF MILLS' DIFFERENTIAL

DEFINITION:

$$\left.\frac{d\Phi(\alpha)}{d\alpha^+}\right|_{\alpha=\tilde{\alpha}} = \lim_{\Delta\alpha \to 0^+} \frac{\Delta(M_{\tilde{\alpha}+\Delta\alpha}) - \Delta(M_{\tilde{\alpha}})}{\Delta\alpha}$$

$$= \lim_{\Delta\alpha \to 0^+} \frac{\Delta(M_{\tilde{\alpha}} + \Delta\alpha(-A)) - \Delta(M_{\tilde{\alpha}})}{\Delta\alpha};$$

(5)

$$\left.\frac{d\Phi(\alpha)}{d\alpha^-}\right|_{\alpha=\tilde{\alpha}} = \lim_{\Delta\alpha \to 0^+} \frac{\Delta(M_{\tilde{\alpha}-\Delta\alpha}) - \Delta(M_{\tilde{\alpha}})}{\Delta\alpha}$$

$$= \lim_{\Delta\alpha \to 0^+} \frac{\Delta(M_{\tilde{\alpha}} + \Delta\alpha A) - \Delta(M_{\tilde{\alpha}})}{\Delta\alpha}.$$

Using Mills' result, we have:
THEOREM 1:

$$\frac{d\Phi(\alpha)}{d\alpha^+}\Big|_{\alpha=\bar{\alpha}} \quad \frac{d\Phi(\alpha)}{d\alpha^-}\Big|_{\alpha=\bar{\alpha}} \text{ exist for } \forall\ \alpha\epsilon R^+$$

and it holds, that

$$\frac{d\Phi(\alpha)}{d\alpha^+}\Big|_{\alpha=\bar{\alpha}} = \Delta((-A), X^0(\bar{\alpha}), Y^0(\bar{\alpha})),\ \bar{\alpha}\epsilon R^+) \tag{6}$$

and

$$\frac{d\Phi(\alpha)}{d\alpha^-}\Big|_{\alpha=\bar{\alpha}} = -\Delta(A, X^0(\bar{\alpha}), Y^0(\bar{\alpha})),\ \bar{\alpha}\epsilon R^+), \tag{7}$$

where $X^0(\bar{\alpha})$, $Y^0(\bar{\alpha})$ are the sets of optimal strategies for the row, column player resp. to the game $M\bar{\alpha}$. The right-hand side terms of (6) and (7) are, apart from the sign of (7), the game values of the constrained games,

$$((-A), X^0(\bar{\alpha}), Y^0(\bar{\alpha})), \tag{8}$$

resp.

$$(A, X^0(\bar{\alpha}), Y^0(\bar{\alpha})), \tag{9}$$

with the matrices $(-A)$, resp. A and the additional restrictions:

$$x\epsilon X^0(\bar{\alpha})$$

$$y\epsilon Y^0(\bar{\alpha}).$$

The terms (6) and (7) need not be the same nor is the one always greater than the other.

REMARK 1: Theorem 1 implies the continuity of function Φ.

LEMMA 1:

$$A > 0 \Rightarrow \frac{d\Phi}{d\alpha^+} < 0, \qquad \forall\ \alpha\epsilon R^+; \tag{10}$$

$$A > 0 \Rightarrow \frac{d\Phi}{d\alpha^-} < 0, \qquad \forall\ \alpha\epsilon R^+; \tag{11}$$

(12) $$A \geqq 0 \Rightarrow \frac{d\Phi}{d\alpha^+} \leqq 0, \qquad \forall\, \alpha \epsilon R^+;$$

(13) $$A \geqq 0 \Rightarrow \frac{d\Phi}{d\alpha^-} \leqq 0, \qquad \forall\, \alpha \epsilon R^+.$$

Lemma 1 is an immediate consequence of Theorem 1.

REMARK 2: If any continuous function satisfies conditions (10), (11) resp., (12), (13) for a certain interval it is monotone decreasing resp. monotone nonincreasing in this interval.

THEOREM 2: If $\exists\ \tilde{\alpha} \epsilon R^+$ such that:

$$\left.\frac{d\Phi(\alpha)}{d\alpha^+}\right|_{\alpha=\tilde{\alpha}} = 0 \ \vee \ \left.\frac{d\Phi(\alpha)}{d\alpha^-}\right|_{\alpha=\tilde{\alpha}} = 0$$

then

(i) there exists a submatrix $\bar{A}$ of A such that

$$\bar{A} = 0;$$

(ii) $$\Phi(\tilde{\alpha}) = \Delta(\bar{B}),$$

where $\bar{B}$ is the corresponding submatrix of B to $\bar{A}$.

PROOF: By Theorem 1 holds:

(14) $$\Delta((-A), X^0(\tilde{\alpha}), Y^0(\tilde{\alpha})) = 0,$$

resp.

(15) $$\Delta(A, X^0(\tilde{\alpha}), Y^0(\tilde{\alpha})) = 0.$$

Without loss of generality let (14) be satisfied and $\tilde{x}, \tilde{y}$ a pair of optimal strategies to it. We now conclude:

a. $$a_{ij} = 0 \qquad \text{for } (i,j) \epsilon K,$$

where

$$K = \{(i,j) \mid \tilde{x}_i \tilde{y}_j > 0, i \epsilon I, j \epsilon J\} \neq \phi.$$

b. The proof follows from the saddlepoint property of the game value $\Phi(\alpha)$ restricting to stochastic vectors corresponding to the matrix $\bar{B}$.

Considering the continuity and monotonicity of function Φ [Remark 1, Remark 2] the following theorem stated by Thompson and Weil in [10] (Th. 4.6) is an immediate consequence of Theorem 2:

COROLLARY 1: (Thompson, Weil)
If $\exists \nu_1, \nu_2 \epsilon R^+ \quad \nu_1 < \nu_2$ such that:

$$\Phi(\alpha) > a \qquad \alpha < \nu_1$$

$$\Phi(\alpha) = a \qquad \nu_1 \leqq \alpha \leqq \nu_2$$

$$\Phi(\alpha) < a \qquad \alpha > \nu_2$$

then

(i) there exists a submatrix $\bar{A}$ of A, such that:

$$\bar{A} = 0;$$

and

(ii) $$\Delta(\bar{B}) = a.$$

Such a decomposition was first stated by Kemeny, Morgenstern, and Thompson in order to prove the existence of subeconomies [5].

COROLLARY 2: There are at most finitely many numbers $\nu \epsilon R^+$, such that

$$\left.\frac{d\Phi(\alpha)}{d\alpha^+}\right|_{\alpha=\nu} = 0 \vee \left.\frac{d\Phi(\alpha)}{d\alpha^-}\right|_{\alpha=\nu} = 0.$$

PROOF: There are at most a finite number of submatrices $\bar{A}$ of A, such that,

$$\bar{A} = 0.$$

REMARK 3: There are at most a finite number of intervals, such that

$$\Phi(\alpha) = \text{const.}$$

Finally we shall study the connections between the differentiability or non-differentiability of function Φ in α_1 resp. α_2, the highest resp. the lowest feasible growth factor (for definitions see the introduction) and the existence of economic solutions to α_1 and α_2.

DIFFERENTIABILITY AND ECONOMIC SOLUTIONS

Let $(\hat{x}, \hat{y})$, resp. $(\bar{x}, \bar{y})$ be any pair of optimal strategies to

$$((-A), X^0(\nu), Y^0(\nu)).$$

resp.

$$(A, X^0(\nu), Y^0(\nu)).$$

THEOREM 3: If

$$\left.\frac{d\Phi(\alpha)}{d\alpha^+}\right|_{\alpha=\nu} \neq \left.\frac{d\Phi(\alpha)}{d\alpha^-}\right|_{\alpha=\nu}$$

where $\nu=\alpha_1 \vee \nu=\alpha_2$ and $\alpha_2 < \alpha_1$ at least one of the pairs of strategies, $(\hat{x}, \hat{y}) \vee (\bar{x}, \bar{y})$ is an economic solution. (For $\nu: \alpha_2 < \nu < \alpha_1$ both of the above mentioned differentials are zero).

PROOF: Without loss of generality, we assume $\nu=\alpha_1$ and

$$\left.\frac{d\Phi(\alpha)}{d\alpha^+}\right|_{\nu=\alpha_1} \neq 0.$$

According to Theorem 1 for the pair $(\hat{x}, \hat{y})$ holds:

(16) $$\hat{x}\epsilon X^0(\alpha_1), \hat{y}\epsilon Y^0(\alpha_1)$$

and

(17) $$\hat{x}A\hat{y} > 0.$$

As α_1 is a feasible growth factor we conclude from (16)

$$\hat{x}M\alpha_1\hat{y}=\hat{x}(B-\alpha_1 A)\hat{y}=0.$$

and therefore from (17)

$$\hat{x}B\hat{y} > 0.$$

Q.E.D.

The following example shows that function Φ may be absolutely smooth in α_1, this means accordance of left- and right-hand side derivatives (although concluded from the usual graphs of Φ e.g., in [3] or [4] they must not be smooth in α_1).

Choose

$$B=\begin{bmatrix}0 & 0 & 2\\ 2 & 0 & 3\\ 1 & 1 & 0\end{bmatrix} \qquad A=\begin{bmatrix}0 & 0 & 2\\ 2 & 0 & 2\\ 2 & 2 & 0\end{bmatrix},$$

then

$$M_\alpha=\begin{bmatrix}0 & 0 & 2-2\alpha\\ 2-2\alpha & 0 & 3-2\alpha\\ 1-2\alpha & 1-2\alpha & 0\end{bmatrix}.$$

For $\alpha = 1-\epsilon$, $\epsilon > 0$ we have

$$M_{1-\epsilon} = \begin{bmatrix} 0 & 0 & 2\epsilon \\ 2\epsilon & 0 & 1+2\epsilon \\ -1+2\epsilon & -1+2\epsilon & 0 \end{bmatrix}.$$

If $\epsilon < \frac{1}{2}$ it follows, $\Delta(M_{1-\epsilon}) = 0$.

For $\alpha = 1$ we have

$$M_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & +1 \\ -1 & -1 & 0 \end{bmatrix}.$$

The optimal strategies are:

$$x = (\xi, 1-\xi, 0), 0 \leqq \xi \leqq 1, y = (\eta, 1-\eta, 0),$$

$$0 \leqq \eta \leqq 1; \qquad \Delta(M_1) = 0$$

and for $\alpha = 1+\epsilon$, $\epsilon > 0$ we have

$$M_{1+\epsilon} = \begin{bmatrix} 0 & 0 & -2\epsilon \\ -2\epsilon & 0 & 1-2\epsilon \\ -1-2e & -1-2\epsilon & 0 \end{bmatrix}.$$

As $\Delta(M_{1+\epsilon}) < 0$, it is obvious that $\alpha_1 = 1$.

In order to evaluate the right-hand side derivative we determine the value of the constrained game

$$((-A); \{(\xi, 1-\xi, 0), 0 \leqq \xi \leqq 1\}, \{(\eta, 1-\eta, 0), 0 \leqq \eta \leqq 1\}).$$

This game value is zero, the left-hand side derivative is zero anyway.

Although the left- and right-hand side derivatives are zero there exists an economic solution to $\alpha = 1$, e.g., $x = (0, 1, 0)$, $y = (1, 0, 0)$.

We think, that the corresponding function of Φ used to prove the existence of solutions to the revised Open Expanding Economy Model [7], [8] can be treated in a similar way which e.g., may help to prove the existence of solutions satisfying Axiom 5. Let us state that the condition (A6) made there is nothing else but a condition on the derivative of the game value function used in [7], [8] for $\alpha >> 0$.

BIBLIOGRAPHY

[1] Burley, S. P., "Calculating von Neumann Trajectories by Simulated Market Adjustments," Zeitschrift für Nationaloekonomie, Suppl. 1, (1971), pp. 131–138.

[2] Charnes A. and W. Cooper, *Management Models and Industrial Applications of Linear Programming* two volumes. (Wiley, New York 1961).

[3] Förstner, K.: Wirtschaftliches Wachstum bei vollständiger Konkurrenz und linearer Technologie, in Operations Research Verfahren vol. 4, pp. 1–258 (Editor: R. Henn), Verlag Anton Hain, Meisenheim a. Glan (1967).

[4] Hamburger, M. J., G. L. Thompson, and R. L. Weil, "Computation of Expansion Rates for the Generalized Von Neumann Model of an Expanding Economy," Econometrica *35*, 452–549 (1967).

[5] Kemeny, J. G., O. Morgenstern, and G. L. Thompson, "A Generalization of the Von Neumann Model of an Expanding Economy," pp. 115–135, Econometrica (1956).

[6] Mills, H. D., "Marginal Values of Matrix Games and Linear Programs," in *Linear Inequalities and Related Systems* (H. W. Kuhn and A. W. Tucker, editors) pp. 183–198 (Princeton, New Jersey 1956).

[7] Morgenstern, O. and G. L. Thompson, "An Open Expanding Economy Model," Nav. Res. Log. Quart. *16*, 443–457 (1969).

[8] Morgenstern, O. and G. L. Thompson, "Further Considerations of An Open Expanding Economy Model," Zeitschrift für Nationaloekonomie, Suppl. 1 pp. 23–30 (1971).

[9] Thompson, G. L., "On the Solution of a Game Theoretic Problem," in *Linear Inequalities and Related Systems* (Editors: H. W. Kuhn and A. W. Tucker) pp. 275–284 (Princeton, New Jersey, 1956).

[10] Thompson, G. L. and R. L. Weil, Jr., "The Roots of Matrix Pencils $(Ay - \lambda By)$: Existence, Calculations and Relations to Game Theory," Management Sciences Research Report No. 172, Carnegie Mellon University, Pittsburgh, Pennsylvania (1969).

[11] Weil, R., "Game Theory and Eigensystems," SIAM Review *X*, 360–367 (1968).

AN EFFICIENT HEURISTIC PROCEDURE FOR THE UNCAPACITATED WAREHOUSE LOCATION PROBLEM

Basheer M. Khumawala

Graduate School of Business Administration
University of North Carolina at Chapel Hill

ABSTRACT

This paper introduces an efficient heuristic procedure for a special class of mixed integer programming problems called the uncapacitated warehouse (plant) location problem. This procedure is derived from the branching decision rules proposed for the branch and bound algorithm by the author in an earlier paper. It can be viewed as tracing a single path of the branch and bound tree (from the initial node to the terminal node), the path being determined by the particular branching decision rule used. Unlike branch and bound the computational efficiency of this procedure is substantially less than linearly related to the number of potential warehouse locations (integer variables) in the problem. Its computational efficiency is tested on problems found in the literature.

1. THE WAREHOUSE LOCATION PROBLEM

The problem dealt with in this paper pertains to a special class of the mixed integer programming problem, generally referred to as the warehouse (plant) location problem.* This problem typically arises in the following way. A number, usually very large, of potential warehouse locations are known to be available where warehouses can be established for performing the distribution function. Associated with each potential location is a fixed cost if a warehouse is opened,† and, associated with an open warehouse are also the costs of its operations. The distribution costs consist of the fixed costs of the open warehouses, the costs of operating these open warehouses, and the transportation costs to and from the open warehouses. The warehouse location problem is then to select the sub-set of the potential warehouse locations to be opened, which meets the constraints of customer's demands and minimizes the distribution costs.

The warehouse location problem is called capacitated or uncapacitated, according to whether or not the capacity constraints on the size of the potential warehouses are imposed. In the capacitated case, the capacities of the potential warehouses are determined a priori, where as in the uncapacitated case, the solution to the problem determines the capacities to be used at the open warehouses. This paper deals with the uncapacitated warehouse location problem and will henceforth be simply referred to as the warehouse location problem. See [2, 4, 6, 11] for solution methods to the capacitated warehouse location problem.

---

*Although the problem is called the warehouse location problem, it occurs in many operational situations: the lock-box problem, the fire (or police) station problem, the detection-station placement problem, the fixed-charge transportation problem, the bill board location problem, the interceptor-missile site problem. These are described by Gray [6] and Ellwein [3].

†The terms "open," "closed," and "free" warehouses will respectively mean a potential warehouse assigned to be used, a potential warehouse assigned not to be used, and a potential warehouse not yet assigned either open or closed.

Mathematically, the warehouse location problem is:*

Minimize
$$Z=\sum_{i,j} C_{ij}X_{ij}+\sum_{i} F_i y_i$$

subject to
$$\sum_{i} X_{ij}=1 \qquad j=1, 2, \ldots n$$

$$0\leq X_{ij}\leq y_i\leq 1 \qquad i=1, 2, \ldots m$$

$$y_i=0 \text{ or } 1 \qquad j=1, 2, \ldots n,$$

where

m = number of potential warehouses,
n = number of customers,
$C_{ij}=t_{ij}D_j$,
t_{ij} = the per unit cost which includes the FOB cost at the warehouse (i), the warehouse operating costs and the transportation costs from the warehouse to the customer (j),
D_j = the demand at customer j,
X_{ij} = fraction of D_j supplied from warehouse i, and
F_i = the fixed cost associated with warehouse i.

The existence of the fixed warehouse costs creates a discontinuity at zero shipments, and prevents the use of standard linear programming (LP) techniques. One approach to solving the warehouse location problem is by complete enumeration of all possible 2^m combinations of the potential warehouse locations. In practice, m is too large to make the combinatorial approach practical. Several heuristic and simulation approaches have been proposed. Prominent among them are [5, 9, 10, 12]. A branch and bound approach (B & B) for finding the optimal solution was first presented by Efroymson and Ray [3] and has been subsequently improved by Khumawala [7]. Other exact algorithms have also been presented by Spielberg [13, 14]. See [8] for a review and comparison of some of these recent approaches.

2. THE OBJECTIVE OF THIS RESEARCH

Although Khumawala reports very efficient results with his algorithm,† it is certainly true that the computational efficiency‡ of B & B will not be independent of the total number of potential warehouses

*The formulation given here is for the simplest case of the warehouse location problem, where each potential warehouse has a fixed cost and a linear variable cost associated with it. However, as shown in [3], the general case of having for each potential warehouse a fixed cost and several linear segments also reduces to the simple case. In the general case, an integer variable is associated with each linear segment of each potential warehouse, and the corresponding fixed cost is obtained by extending the linear segment until it intersects the cost axis. Thus, each potential warehouse is viewed as consisting of several warehouses. The computation time, however, increases more rapidly with the number of potential warehouses, than with the number of linear segments.

†Khumawala [7] reports an average time of less than 10 sec on IBM 360/65 for solving the Kuehn and Hamburger [9] problems, which consist of 25 potential warehouses and 50 customers.

‡The criteria for measuring the computation efficiency is the computation time and computer storage required to solve a given warehouse location problem.

in the problem. In fact, this relationship is exponential. This means that problems involving very large number of potential warehouses* cannot be solved efficiently. The objective of our research was to overcome this difficulty.

We have developed an efficient heuristic procedure whose computational efficiency as related to the number of potential warehouses is substantially less than linear and at the same time it provides very good solutions. The heuristic procedure consists of solving the warehouse location problem by a set of simple and computationally inexpensive heuristic rules. It is this aspect of our heuristic procedure that makes it superior to other heuristic procedures, which are "specialized" to particular types of problems; i.e., some heuristics will work well on some types of location problems and may not work well on others. By work well we are referring to the problem of cost. Our individual heuristic rules will undoubtedly be specialized for particular types of problems, but when they are combined together, they yield excellent results.

The heuristic rules can be viewed as tracing a single path of the "tree," resulting from the application of the B & B algorithm to the warehouse location problem. The path is determined by the particular branching decision rule† used. The eight branching decision rules of Khumawala [7], therefore give eight different solutions to the problem.‡ The best (minimum in this case) of these eight solutions is taken as the "solution" to the given warehouse location problem.**

There are two main advantages of this heuristic procedure:

(1) In practice management is not interested in global optimal solutions to warehouse (plant) location problems. Rather, management is interested in some "least cost" alternatives with an assurance that the solutions of these alternatives are not very far from the optimum. The heuristic procedure does precisely that. It finds several "good" solutions to the given problem, each solution being obtained by the use of the corresponding branching decision rule.

(2) If a guaranteed optimal solution is desired, one can now apply the B & B algorithm of [7] using the heuristic solution as a starting upper bound to the problem. This should improve the computational efficiency of the exact solution method.

We begin by first reviewing briefly that portion of the B & B algorithm which is relevant to our discussion in this paper. The heuristic rules are then described, which are followed by the computational results.

3. THE BRANCH AND BOUND ALGORITHM

Problem I is first solved as a linear program (without the integer restrictions on the Y's). Let Z_0 be the solution. If all the Y's are integer, than the problem is solved. Otherwise Z_0 becomes a lower bound to problem I. Now, select a fractional Y_k and solve two additional problems with constraints

---

*For the sake of convenience, we shall refer to warehouse location problems involving a large number of potential warehouses as simply large warehouse location problems.

† At each successive stage, the B & B algorithm requires the selection of a free warehouse from the set of free warehouses at that stage, to be constrained open and closed. The rule by which the free warehouse is selected is called the branching decision rule.

‡ It is not necessary that all eight heuristic rules be used to determine the solution to the given problem. However, since each of these rules takes almost very little computer time to obtain its respective solution, the use of all eight is included in our procedure. It should also be noted that additional branching decision rules can be developed which in turn would generate additional heuristic rules.

**We shall henceforth refer to the solution obtained in this manner as the heuristic solution. This heuristic solution of course, cannot be guaranteed as optimal.

$Y_k=0$ (free warehouse k constrained closed) and $Y_k=1$ (free warehouse k constrained open), respectively. In a tree representation of this procedure, we say that two branches emanate from the node at which the solution is Z_0. If Z_1 and Z_2 are the two new solutions (clearly $Z_1 \geq Z_0$ and $Z_2 \geq Z_0$), then $\bar{Z}=\min(Z_1, Z_2)$ is a new improved lower bound to problem I. This procedure of branching and bounding continues till the current upper bound equals the current lower bound.

Certain simplifications at each node have been suggested in [3], which are repeated here for convenience. These simplifications have the advantage of reducing the number of branches emanating from the node that need to be evaluated for guaranteeing optimality.

SIMPLIFICATIONS*

1. The first simplification determines a minimum bound for opening a warehouse. If this bound is positive the warehouse will be fixed open. Mathematically, this is, for $i \epsilon k_2$, $j \epsilon P_i$, compute

$$\Delta_i=\sum_{j\epsilon P_i}\left\{\min_{\substack{k\epsilon K_1 \cup K_2 \\ k\epsilon N_j \\ k\neq i}}\left[\max\left(C_{kj}-C_{ij}, 0\right)\right]\right\}-F_i.$$

If $\Delta_i > 0$, then $y_i=1$ (warehouse i is fixed open at this node) for all branches emanating from that node.

2. The second simplification provides a means for reducing n_i as follows: for $i \epsilon K_2$, $j \epsilon P_i$, if $\underset{k\epsilon K_1 \cap N_j}{\text{Min}}$ $(C_{kj}-C_{ij}) < 0$, then n_i is reduced by one. If the inequality holds for all $j \epsilon P_i$, then n_i will be zero and $Y_i=0$ for all branches emanating from that node.

3. The third simplification determines a maximum bound on the cost reduction for opening a warehouse.

Mathematically, this is, for $i \epsilon K_2$, $j \epsilon P_i$

$$\Omega_i=\Sigma_{j\epsilon P_i}\{\min_{\substack{k\epsilon K_1 \\ k\epsilon N_j}}[\max(c_{kj}-c_{ij}, 0)]\}-F_i.$$

If $\Omega_i < 0$, then $y_i = 0$ (warehouse i is fixed closed at this node) for all branches emanating from that node.

In [7], the information provided by these simplifications is used to further advantage in two ways:

(i) in efficiently solving the linear program at each node.

(ii) in determining efficient rules for selecting the free warehouse k (corresponding to the fractional y_k) to be constrained closed ($y_k = 0$) and open ($y_k = 1$) at each node.

The following are the eight branching decision rules proposed in [7], which form the basis of the heuristic rules to be presented here.

1. Largest Delta Rule
2. Largest Omega Rule
3. Largest Demand Rule
4. Largest Y Rule
5. Smallest Delta Rule
6. Smallest Omega Rule
7. Smallest Demand Rule
8. Smallest Y Rule

---

*The mathematical notations of [3] are used in this paper. Thus, N_j denotes the set of warehouses that can supply the jth customer; P_i is the set of customers that can be supplied by the ith warehouse; n_i is the number of elements in the set P_i; K_0, K_1, and K_2, respectively, denote the set of warehouses that are fixed closed, fixed open, and free at the node.

The largest delta branching decision rule selects the free warehouse, which has the largest Δ from the set of free warehouses at the node having negative Δ. On the other hand, the smallest delta rule selects the free warehouse which has the smallest Δ. The rationale here is that, if $\Delta_i > 0$, then the warehouse i is fixed open at this node, according to simplification one. However, from the warehouse whose Δ are negative, those having large Δ values are likely to be open at the terminal node reached from this node. The other branching decision rules can be explained in a similar manner.

4. THE HEURISTIC PROCEDURE

The main difficulty with B & B is computational, which has two dimensions—computer time and computer storage. This is because the number of nodes in the B & B algorithm that need to be evaluated to reach an optimal solution has an exponential relationship with the total number of integer variables in the problem.* An increase in the number of nodes is accompanied by an increase in computation time as well as computer storage; an LP problem is required to be solved at each node, and all nodes whose LP solutions are less than the current upper bound have to be stored. Thus, the B & B algorithm for the warehouse location problem even in its present efficient form (as in [7]) may not be satisfactory for very large problems.

The heuristic procedure, which we shall now describe, overcomes this difficulty. The relationship of the computational efficiency of this procedure to the number of integer variables in the problem is substantially less than linear, and at the same time it provides quite good solutions. This procedure is based upon the branching decision rules listed earlier. Corresponding to the eight branching decision rules, eight heuristic rules are first derived. These heuristic rules form the nucleus of our procedure. We first define what we shall mean by a "preferred path." This will then be conveniently used to describe the eight heuristic rules.

Recall that, in order to apply the B & B algorithm, a free warehouse k (fractional y_k) is selected from the set of free warehouses at each node. This free warehouse is constrained both open ($y_k = 1$) and closed ($y_k = 0$), which results in two additional nodes. The two emanating branches along which the free warehouse is constrained open and closed are respectively called the open and the closed branch. The eight branching decision rules were developed to make a judicious selection of the free warehouses at each node. These rules are to two types:

(1) Those branching decision rules that are based on the reasoning that the free warehouse selected will most likely be open in the best (minimum) solution reached from the current node.† The branching decision rules 1 to 4 are of this type.

(2) Those branching decision rules that are based on the reasoning that the free warehouse selected will most likely be closed in the best (minimum) solution reached from the current node. The branching decision rules 5 to 8 are of this type.

The word "preference" will be used here to mean the motivation on which the free warehouse is selected. Thus, the open branch is the preferred branch for branching decision rules of the first

*Although an exact relationship between the number of nodes that need to be evaluated and the number of integer variables in the problem cannot be known, we can say that the maximum number of terminal nodes will be $2^m - 1$, where m is the number of integer variables; we eliminate the case of no warehouses in the system.

†Included in this rationale is also the reasoning that the selection of this free warehouse will make the current node converge rapidly to its terminal nodes.

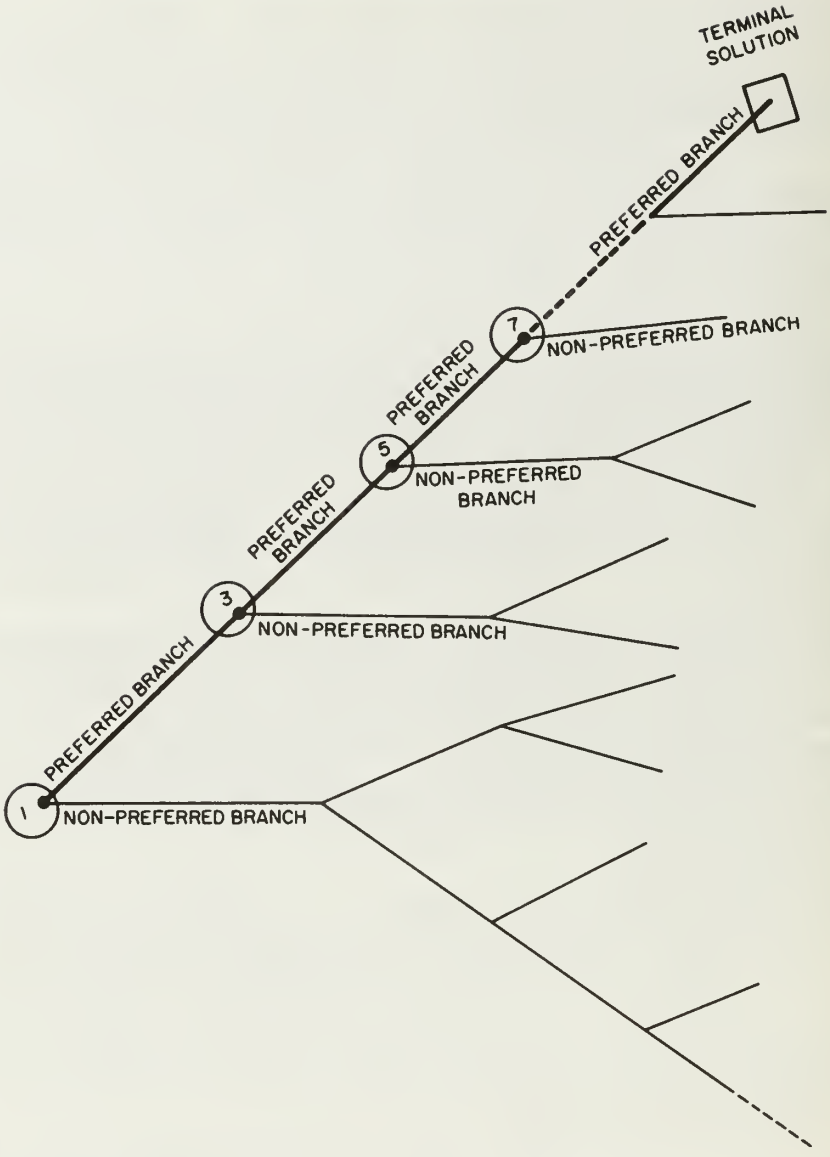


FIGURE 1. The preferred path.

type. The closed branch, on the other hand, is the preferred branch for branching decision rules of the second type.

As the branching decision rule is applied in the B & B algorithm starting from the initial node (node 1 in Figure 1), this means that we expect the best solution to be along the preferred branch. If the node resulting from the preferred branch (node 3 in Figure 1) is not terminal, branching continues further. If branching continues from node 3, we would again expect the preferred branch from this node to yield the best solution and so on till the terminal node is reached. The path along the preferred branches from the initial node to the terminal node is defined as the preferred path. In the figure, it is given by 1–3–5–7 . . ..

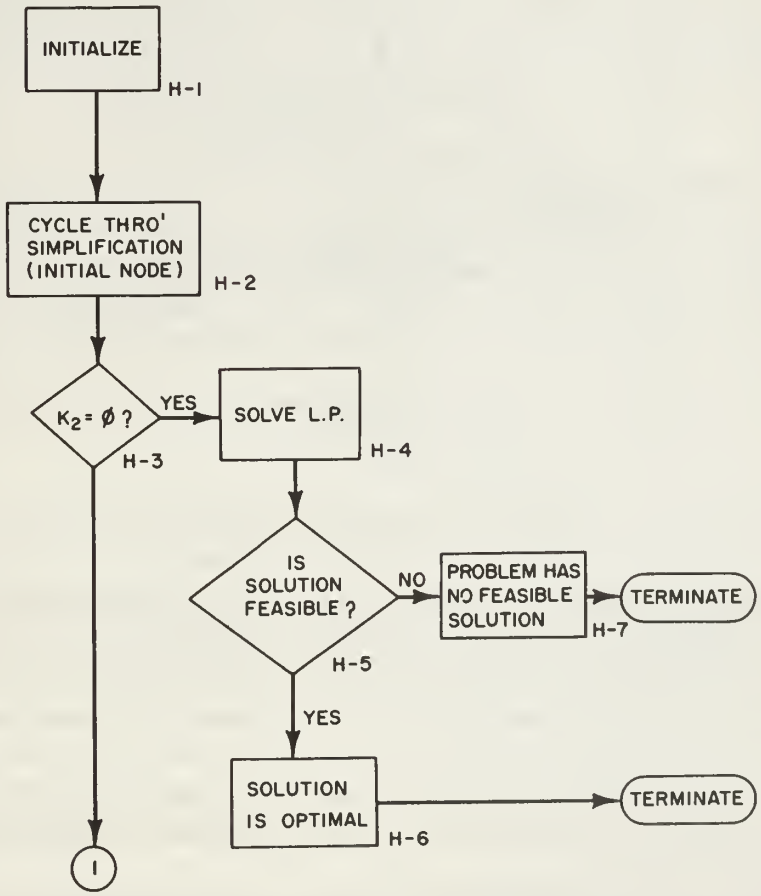
The heuristic rule can now be simply stated as: "Trace the preferred path which would result from the application of the particular branching decision rule." With eight branching decision rules,

we therefore have eight heuristic rules. Similar to the branching decision rules, the heuristic rules are also classified into two types. In type 1, the preferred path consists of open branches, whereas, in type 2, the preferred path consists of closed branches.

It is interesting to note that a type 1 rule is equivalent to the add heuristic of Kuehn and Hamburger [9], where a potential warehouse is added at each stage. On the other hand, a type 2 rule is equivalent to the drop heuristic of Feldman, Lehrer, and Ray [5], where a potential warehouse is dropped at each stage.

The complete heuristic procedure is described by the flow diagram in Figure 2. The individual steps in this diagram, are numbered with prefix *H*. The reader is referred to the flow diagrams of the B & B procedure and the simplification cycle given in [7]. These latter diagrams are numbered with prefix *M* and *S*, respectively. We make use of these diagrams for describing the individual steps of Figure 2.

Steps *H*−1 and *H*−2 are identical to steps *M*−1 and *M*−2. At step *H*−1, the sets N_j, P_i, K_0, K_1, and K_2 are initialized for the first node. Step *H*−2 then cycles through the simplifications (steps *S*−1 through *S*−12) until no more free warehouses can be fixed open or closed. If at this stage, all warehouses are assigned open and closed, i.e., the set K_2 is empty, the LP is solved (*H*−4), and the procedure terminates. If the LP solution is infeasible,† the given warehouse location problem has no



† Infeasible conditions will actually be determined when computing the Δ_{ij}'s (step *S*−1) during the simplification cycle.

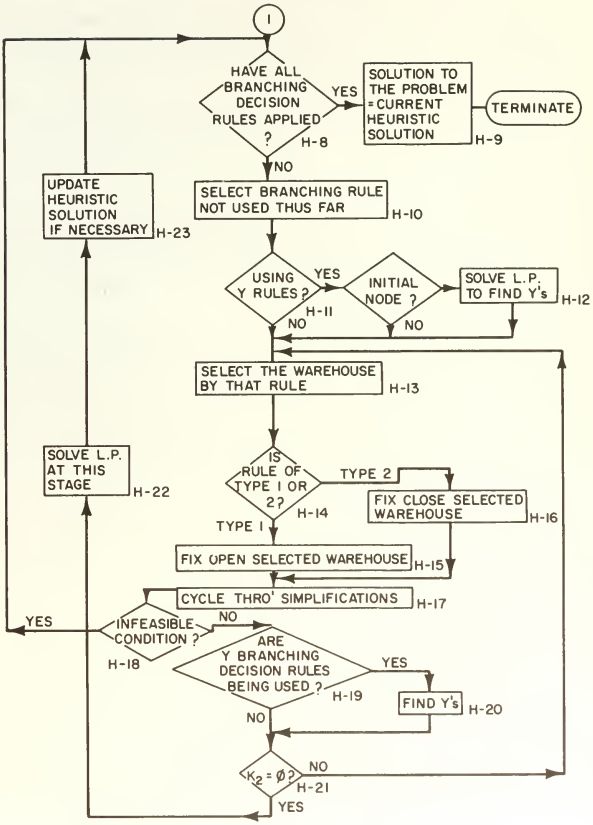


FIGURE 2. Flow chart of the heuristic procedure.

feasible solution, ($H-7$). Otherwise, the LP solution is the guaranteed optimal solution for the warehouse location problem ($H-6$).

However, if all warehouses are not assigned fixed open or closed at this node, i.e., the set K_2 is not empty (this will be the usual case), the procedure continues. A branching decision rule is selected ($H-10$) which in turn selects the free warehouse from the set K_2 to be constrained ($H-13$). The information* required to do the selection of the free warehouse is already available for all the branching decision rules except for the Y rules (the largest Y and the smallest Y). Thus, if the Y branching decision

*This information is available from step $H-2$, when the procedure cycles through the simplifications. The delta, the omega, and the demands are all calculated as part of the simplifications.

rules are employed, the Y's are to be calculated by solving the LP problem at the initial node (step $H-12$).

If the branching decision rule to be used is of the first type, the selected warehouse is constrained open ($H-15$). However, if it is of the second type, the selected warehouse is constrained closed ($H-16$). In either case, the procedure cycles through the simplifications at this node until no further improvements are realized ($H-19$). It is possible for infeasible conditions to arise when a free warehouse is constrained closed by either a type 2 branching decision rule (step $H-18$) or during the simplification cycle. This will happen if the free warehouse which gets closed happens to be the only supplier of a customer. If this happens, the procedure returns to step $H-8$ to continue the procedure using another branching decision rule. However, if infeasible conditions do not exist, the procedure continues further with the same branching decision rule.

Again, only in the case of the Y branching decision rules, it is necessary at this stage to determine the Y's by solving the LP problem at this node ($H-20$). If all the warehouses are not yet assigned fixed open or closed, the procedure is repeated by selecting now another warehouse from the set K_2 ($H-13$). The steps $H-13$ through $H-21$ are cycled until the set K_2 is empty, i.e., now the terminal node is reached, where all the warehouses are either open or closed. The LP problem is solved at the terminal node ($H-22$).* If this solution is better (less) than the heuristic solution obtained thus far, it is updated, i.e., replaced by the better solution ($H-23$).†

If all the branching decision rules are not yet applied to the given problem, another rule is selected and the procedure is cycled through steps $H-10$ to $H-23$. Otherwise, the heuristic algorithm terminates with the heuristic solution as the "solution" to the warehouse location problem.

Illustration

In order to illustrate the application of the heuristic procedure, we shall use the small illustrative warehouse location problem of [7], which we give here in Table 1. This problem is used merely to illustrate the application of the procedure and not to test its computational efficiency. Its efficiency is tested on larger problems, which are discussed later. The B & B tree given in Figure 3 resulted from the application of the largest omega branching decision rule ([7], p. 25). The optimal solution is 1235 and it occurs at node 3.

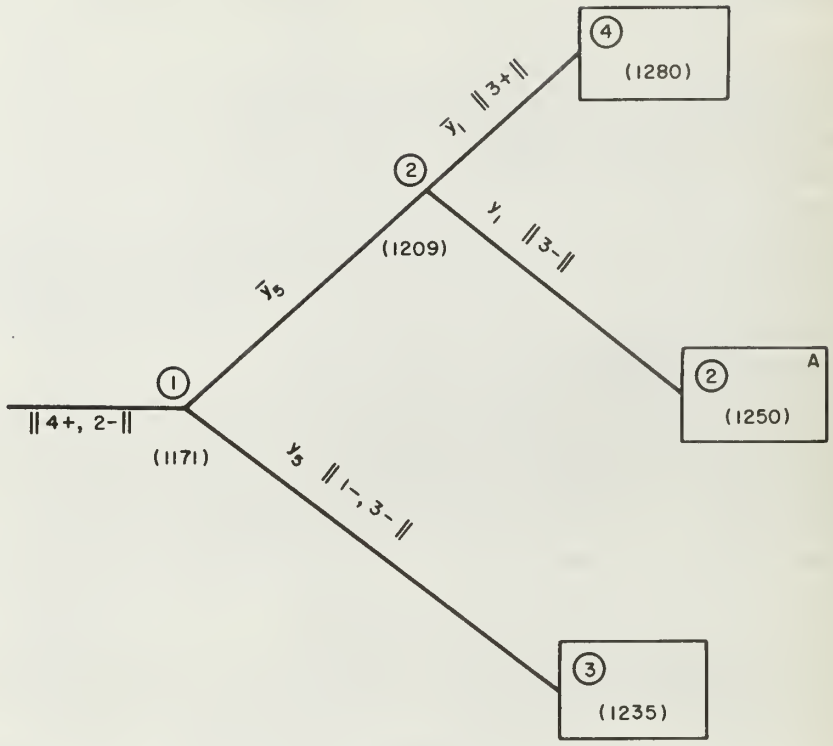
TABLE 1. *Illustrative Problem*

Fixed warehouse costs (F_i)	Warehouse	Customers (C_{ij})							
		1	2	3	4	5	6	7	8
\$100	1	\$120	\$180	\$100	Ⓛ	\$60	Ⓛ	\$180	Ⓛ
70		210	Ⓛ	150	\$240	55	\$210	110	\$165
60	3	180	190	110	195	50	Ⓛ	Ⓛ	195
110	4	210	190	150	180	65	120	160	120
80	5	170	150	110	150	70	195	200	Ⓛ
Customer demands (units)		10	10	10	10	10	10	10	10

Ⓛ Indicates a prohibitive route.

*This solution is already available in the case when the Y branching decision rules are used.

†At the start of the algorithm, the heuristic solution is the starting value of the upper bound (= infinity, a very large quantity in the computer). The heuristic solution is improved as the branching decision rules are applied.



KEY

□	TERMINAL NODE
○	NODE NUMBER
( )	SOLUTION OF LINEAR PROGRAM
y_i	SELECTED WAREHOUSE i CONSTRAINED OPEN
$\bar{y}_i$	SELECTED WAREHOUSE i CONSTRAINED CLOSED
|| r+, S- ||	DUE TO SIMPLIFICATIONS, WAREHOUSE r IS FIXED OPEN AND WAREHOUSE S IS FIXED CLOSED

FIGURE 3. Branch and bound tree for illustrative problem.

If the heuristic procedure was used to solve the illustrative problem, the optimal solution (1235) would still have been found. The preferred path under the largest omega heuristic rule is 1–3, the branch along which warehouse 5 is fixed open. The procedure under the largest omega heuristic rule would simply evaluate nodes 1 and 3 and the heuristic solution would be the LP solution found at node 3, i.e., 1235. (It does not appear necessary to also show the different solutions which would be reached when other heuristic rules are also applied to this problem).

5. COMPUTATIONAL RESULTS

In order to test the efficiency of the procedure outlined in this paper, we applied it to solve the following warehouse location problems which have appeared in the literature. The results are given in Tables 2–4.

(1) The three sets of Kuehn and Hamburger [9] problems consisting of 25 potential warehouses

and 50 customers; each set has four levels of fixed costs. The three sets correspond to the cases in which the factories are at (a) Indianapolis, (b) Jacksonville, Florida, and (c) Baltimore and Indianapolis. *Sã* [11] added one more set by considering the case of having a factory at Indianapolis, but with Indianapolis not serving as a potential warehouse. The results from the application of our procedure on these problems are given in Tables 2 and 3.

TABLE 2. *Computational results*[a]

Branching decision rule	Fixed warehouse costs							
	25 × 50 Indianapolis				25 × 50 Jacksonville			
	($)7,500	12,500	17,500	25,000	7,500	12,500	17,500	25,000
Largest Delta	[b] 796,646	859,919	899,905	937,651	1,093,837	1,151,424	1,197,251	1,247,753
Largest Demand	797,655	855,970	914,582	979,209	1,094,047	1,151,645	1,231,298	1,298,983
Largest Omega	797,507	[b] 854,702	895,025	[b] 928,940	[b] 1,092,914	1,147,458	1,201,051	[b] 1,244,986
Largest *Y*	797,507	[b] 854,702	902,411	935,940	1,096,430	1,156,461	1,197,538	1,261,675
Smallest Delta	[b] 796,646	855,465	[b] 893,780	934,585	1,093,837	1,147,458	1,189,777	[b] 1,244,986
Smallest Demand	801,586	862,446	898,550	[b] 928,940	1,095,756	1,148,950	1,207,545	1,268,080
Smallest Omega	797,507	858,018	895,025	[b] 928,940	1,093,727	1,147,458	1,206,529	1,251,529
Smallest *Y*	797,507	[b] 854,702	894,006	934,585	1,094,943	1,147,458	1,192,486	1,252,126
Heuristic Solution	[b] 796,646	[b] 854,702	[b] 893,780	[b] 928,940	[b] 1,092,914	1,147,458	1,189,777	[b] 1,244,986
Optimal Solution	796,646	854,702	893,780	928,940	1,092,914	1,145,925	1,188,244	1,244,986

[a] The results given in these tables do not exactly match those in [5] and [9], because of slight discrepancies in the data used.

[b] Denotes optimal solution.

TABLE 3. *Computational results*

Branching decision rule	Fixed warehouse costs							
	26 × 50 Indianapolis & Baltimore				24 × 50 Indianapolis (No Warehouse at Indianapolis)			
	($)7,500	12,500	17,500	25,000	7,500	12,500	17,500	25,000
Largest Delta	[a] 614,546	666,667	692,424	727,951	[a] 806,143	875,504	920,491	976,925
Largest Demand	615,555	[a] 659,981	[a] 690,744	734,974	[a] 806,143	877,063	945,815	1,022,440
Largest Omega	615,406	[a] 659,981	690,970	[a] 724,885	[a] 806,143	[a] 870,790	921,010	[a] 970,444
Largest *Y*	615,406	[a] 659,981	690,970	725,404	808,665	875,796	936,143	992,172
Smallest Delta	[a] 614,546	660,744	[a] 690,744	[a] 724,885	[a] 806,143	[a] 870,790	[a] 919,991	974,471
Smallest Demand	618,066	665,089	707,387	[a] 724,885	814,318	876,889	921,010	985,451
Smallest Omega	615,406	[a] 659,981	690,970	[a] 724,885	[a] 806,143	875,290	921,010	982,744
Smallest *Y*	615,406	[a] 659,981	690,970	[a] 724,885	808,665	875,796	923,603	978,083
Heuristic solution	[a] 614,546	[a] 659,981	[a] 690,744	[a] 724,885	[a] 806,143	[a] 870,790	[a] 919,991	[a] 970,444
Optimal solution	614,546	659,981	690,744	724,885	806,143	870,790	919,991	970,444

[a] Denotes optimal solution.

TABLE 4.—*Computational results*

Branching decision rule	49 × 200 (4 Plants)		100 × 200 (4 Plants)	
	Curve *A*	Curve *B*	Curve *A*	Curve *B*
Largest Delta	719,221	719,845	717,307	Infeasible
Largest Demand	707,534	743,140	705,625	Infeasible
Largest Omega	[a] 705,330	721,024	[a] 694,332	Infeasible
Largest *Y*	706,614	722,867	695,077	737,447
Smallest Delta	706,614	715,967	695,092	713,978
Smallest Demand	714,741	724,123	725,784	727,002
Smallest Omega	[a] 705,330	728,538	[a] 694,332	739,685
Smallest *Y*	708,898	719,914	711,913	718,612
Heuristic solution	[a] 705,330	715,967	[a] 694,332	713,978
Optimal solution	705,330	715,328	694,332	N/A

[a] Denotes optimal solution.
N/A Not Available at the time this report was prepared.

(2) The test problems used by Feldman, Lehrer, and Ray [4], consist of factories at four locations, 49 potential warehouses and 200 customers. They solved the problem for two types of cost curves—linear and concave; the following tabulation gives their cost curve data.

Curve *A*		Curve *B*	
Volume	Cost	Volume	Cost
0	0	0	7,140
5,000	50,000		
15,000	100,000	10,000	71,440

Table 4 shows the results obtained when our procedure was used to solve these problems.

Following these results we decided to test the procedure on a much larger problem. We selected 51 additional warehouse locations from the 200 cities given in Feldman, Lehrer, and Ray ([5], p. 676). This gave us a warehouse location problem consisting of 4 factories, 100 potential warehouses, and 200 customers. We used the same two cost curves *A* and *B* for the larger problem.* The results for this problem are also given in Table 4.

The computer times (CPU times) for obtaining the solutions by the individual heuristic rules were in the order of less than 1 sec for 25 × 50 problems, 2–5 sec for 49 × 200 problems, and 5–10 sec for 100 × 200 problems on IBM 360/75. Computer storage on the other hand is merely in direct proportion to the size of the problems at hand, as no nodes are required to be stored.

*For constructing this larger problem the grid system was used to measure the airline distance between cities. This was necessary because the original data of [5] does not have distance between cities in miles. It consists of transportation costs for every customer from its 15 closest warehouses. With the additional 51 warehouse locations the set of 15 closest warehouses for each customer had to be determined again.

The tables show that the heuristic procedure performs extremely well. It gives optimal solutions in most cases tested and in the few exceptional cases, the deviation from the optimal solution is very small. Because the individual heuristic rules take very small computation time to reach their solutions, we recommend the use of all the eight rules for determining the final heuristic solutions. As noted earlier, this of course is not necessary. Analyzing the results in the tables, it can be seen that some individual heuristic rules do perform better than others. One could therefore determine the relative effectiveness of these individual heuristic rules and include in the algorithm only the most promising ones.

REFERENCES

[1] Atkins, R. J. and R. H. Shriver, "New Approach to Facilities Location," Harvard Business Review (May–June 1968).

[2] Davis, P. S. and T. L. Ray, "A Branch-Bound Algorithm for the Capacitated Facilities Location Problem," Nav. Res. Log. Quart. *16*, 331–344 (Sept. 1969).

[3] Efroymson, M. A. and T. L. Ray, "A Branch-Bound Algorithm for Plant Location," Operations Research, Vol. 14 (May–June 1966).

[4] Ellwein, L. B., "Fixed Charge Location-Allocation Problems with Capacity and Configuration Constraints," Technical Report No. 70–2, Department of Industrial Engineering, Stanford University (Aug. 1970).

[5] Feldman, E., F. A. Lehrer, and T. L. Ray, "Warehouse Locations Under Continuous Economies of Scale," Management Science, Vol. 12 (May 1966).

[6] Gray, P., "Mixed Integer Programming Algorithms for Site Selection and Other Fixed Charge Problems Having Capacity Constraints," Department of O. R. and Statistics, Stanford University, Technical Report No. 101 (Nov. 1967).

[7] Khumawala, B. M., "An Efficient Branch and Bound Algorithm for the Warehouse Location Problem," Management Science, Vol. 18, No. 12 (Aug. 1972). Also, in Krannert School of Industrial Administration Paper No. 294, Purdue University (Dec. 1970).

[8] Khumawala, B. M. and D. C. Whybark, "A Comparison of Some Recent Warehouse Location Techniques," The Logistics Review, Vol. 7, No. 3, (Spring 1971).

[9] Kuehn, A. A. and M. J. Hamburger, "A Heuristic Program for Locating Warehouses," Management Science, Vol. 9 (July 1963).

[10] Manne, A. S., "Plant Location Under Economies of Scale—Decentralization and Computations," Management Science, Vol. 11 (Nov. 1964).

[11] Sā, G., "Branch-and-Bound and Approximate Solutions to the Capacitated Plant-Location Problem," Operations Research, Vol. 17 (Nov.–Dec. 1969).

[12] Shycon, H. N. and R. B. Maffei, "Simulation—Tool for Better Distribution," Harvard Business Review (Nov.–Dec. 1960).

[13] Spielberg, K., "An Algorithm for the Simple Plant Location Problem with Some Side Conditions," Operations Research, Vol. 17 (Jan.–Feb. 1969).

[14] Spielberg, K., "Plant Location with Generalized Search Origin," Management Science (Nov. 1969).

THE INITIAL PROVISIONING DECISION FOR INSURANCE TYPE ITEMS*

Robert W. Burton

and

Stratton C. Jaquette

Science Applications, Inc.
Sunnyvale, California

ABSTRACT

Initial provisioning decisions (inventory stocking requirements) for low demand items often have to be made without much knowledge of what future demand rates will be. When the nature of an item is such that little demand for it is expected, the problem of whether to stock initially or risk not stocking the item is most critical. This report discusses this problem and presents decision procedures which can be used to handle this aspect of initial provisioning. The procedures relate an item's provisioning desirability to its provisioning characteristics, such as expected cost, expected resupply time, current information on its likely demand rate, and to an overall operating policy or criterion. The criterion function measures the total system degredation as a function of the events of having items out of stock when demand occurs. Several different policy functions are discussed and the provisioning decision rules which apply to each are presented. Demand rate information is handled through a Bayesian type approach. The decision rules presented in this report can be utilized to either determine stocking requirements within a budgetary constraint, or determine the relative stocking desirability on an item-by-item basis.

1.0 INTRODUCTION

A difficult decision facing the manager in the initial inventory provisioning for complex systems is specifying what inventory protection to provide for items with very low projected demand rates. The manager's initial reaction might be to provide no backup stock for these items. However, other factors such as an item's essentiality to the system, its cost, its resupply time, and/or the uncertainty in original demand estimates might sway him toward providing at least minimal inventory protection. Items or components of this type which he decides to stock can be thought of as "insurance" items.

It should be noted that the decision to stock expensive items in this category can be especially costly. Overstocking of high demand items, although it may be expensive in terms of high cost of current inventory, is self-correcting in a relatively short time. Overstocking of low demand items is not immediately self-correcting, since the misallocation lasts for a much longer time. There is also a high probability that the overstock of a low demand item will never be used and will become obsolete.

---

*This work was conducted for the Naval Supply Systems Command under Office of Naval Research contract N00014–70–C–0119 while the authors were with DECISION Studies Group.

This paper presents decision procedures and a methodology which can be used in initial provisioning to determine which of a group of insurance type items to stock. It assumes that after the provisioning for high demand items, a fund of money is available to stock some but not all of the low demand items. In particular, the subject of this paper is the formulation and analysis of a "stock—no stock" initial provisioning model, where either no spares or one spare is stocked.

Many difficult decisions are inherent in determining what level of system backup stock is required when considering initial provisioning. Three important questions which arise are: (a) how vital or essential is the item, (b) what will the replacement (demand) rate be, and (c) how much money is available for backup stock? This paper is concerned with replacement factor forecasting and budget constraints as they affect the question of initial provisioning, but is not concerned with an investigation of essentiality.

As part of this study, several methods for determining which items to stock and which not to stock are derived. Each method is related to a particular inventory performance objective. The methods link the characteristics of the item to the objective in a consistent and comprehensive way to allow the analyst to make more rational decisions. The characteristics of the item which affect the decision are the following:

- cost estimates of item, both initial and replacement,
- item essentiality,
- best replacement factor for the item,
- uncertainty associated with this replacement factor,
- time required to resupply the item if not stocked,
- criterion or objective for the entire provisioning system, and
- total budget available.

In section 2 the demand and replacement mechanism as it has been modeled for this study is discussed. This includes the assumptions imposed to obtain a workable model, the treatment of the problem of unknown or vaguely known replacement factors, and some discussion of the behavior of backorders for the model presented.

Section 3 is devoted to a discussion of overall system objectives. The difficulty in formulating a specific solution procedure, as opposed to a general methodology, is determining the user's utility of various events. For example, how are the following events rated: three items short for 1 week, one short for 3 weeks, one short at three separate times for a week? Four possible criteria are formulated and discussed.

The methodology to determine which items to stock and which not to stock, a budget allocation problem, is given in section 4. Also discussed are specific applications to the four system objective criteria discussed in section 3. This section closes with an investigation of the behavior of these four criteria and how they compare to each other. A brief discussion of an alternative approach to initial provisioning is given in section 5. In this case the problem is cast as a system performance problem, rather than a budget problem.

Section 6 contains technical derivations that are of little interest to the casual reader in understanding the results of this report. They are, however, important as background and as verification of the results.

2.0 RELATION OF REPLACEMENT FACTORS TO BACKORDERS

In this section we discuss the mathematical concepts used to measure the efficiency of the supply

system in handling insurance type items. We also define some of the random quantities that we need in the analysis and optimization which follows. Since the demand is assumed to be parameterized by replacement rate factors, we also must face the problem of estimating these replacement factors within the limits of existing and reasonable data bases.

2.1 The Replacement Model

The model assumes that the time between demands for any given item is exponentially distributed. This assumption has the benefit of greatly simplifying the analysis and is justifiable for the items in question. This is a common assumption for demand or failure processes and closely reflects the demand or failure behavior in many actual situations (see Sherbrooke [7]). There are no aspects of the current problem that would indicate that a different demand distribution would be appropriate. Indeed, with low demand rates the forthcoming data on demand will likely be insufficient to prove or disprove this or any other such assumption in any practical or statistically valid sense.

Defining $1/\lambda$ as the mean time before replacement for an item, we can easily write the probability distribution of D_t, the number of demands that occur up to time t as

$$Pr\{D_t = k\} = (\lambda t)^k e^{-\lambda t}/k!,$$

i.e., D_t is a Poisson process with parameter λ.

The random variable of interest is the number of backorders. If we let T denote the expected resupply time and s denote the number of items initially stocked for the item (in this case s is zero or one), we can write the probability distribution for the number of backorders, B, at some arbitrary point in time under steady state conditions as follows:

$$Pr\{B=k\} = \begin{cases} (\lambda T)^{k+s} e^{-\lambda T}/(k+s)! & k \geq 1 \\ \sum_{i=0}^{s} (\lambda T)^i e^{-\lambda T}/i! & k=0. \end{cases}$$

In section 6 we discuss the efficacy of using the expected resupply time, T, when the actual time is variable and/or unknown. We also need the expected number of backorders at an arbitrary point in time. Assuming a constant stocking policy and a known replacement rate, we can easily find the expected backorders for the stock levels of interest.

$$E[B] = \begin{cases} \lambda T & \text{is } s=0 \\ \lambda T - [1 - e^{-\lambda T}] & \text{is } s=1 \\ \lambda T - [1 - e^{-\lambda T}] - [1 - e^{-\lambda T} - \lambda T e^{-\lambda T}] & \text{if } s=2. \end{cases}$$

A derivation of these expressions is contained in section 6.

2.2 Failure Model with Unknown Failure Rate

In general, we do not know the value of λ associated with an item. We therefore assume that instead of a point estimate for λ, we have an a priori distribution for λ, i.e., a Bayes estimate. (For a

discussion of Bayesian estimation methods, see Good [4].) In particular we are given a probability density function $g(\lambda)$ which defines the likely values of λ in the following manner:

$$Pr\{\lambda \leq x_0\} = \int_0^{x_0} g(x)dx.$$

The a priori distribution, $g(x)$, is often found by first choosing a family of distributions that are easily parameterized and intuitively reasonable and then choosing that member of the family that best satisfies the intuition and the data available. The data points may be experimental or historical.

For the specific case of the Navy Supply System, considerable data are available for estimating the demand rate, i.e., selecting a specific a priori distribution. The type of data available is in the form of tables of historically experienced replacement factors grouped by categories of equipment types. Also available are engineering estimates and/or the manufacturer's estimates of the replacement rates. This information is currently used to establish a Best Replacement Factor (BFR) for a new item by using the median of the historical replacement rates for that general category of equipment. The data can be used more effectively, however, if the entire table of values is used along with the engineering estimates to generate an a priori distribution for the replacement rate.

2.2.1 The Gamma Distribution

The gamma family of densities is particularly well suited as a candidate for an a priori distribution of failure rates. First it allows λ to range over $(0, \infty)$, as we intuitively expect; second the parameters which define the family admit many different acceptable shapes of densities (see Figure 1a), and finally it is very easy to use analytically and practically.

The gamma family is parameterized by a pair of numbers (α, p). The density function for the replacement rate λ associated with the pair (α, p) is given by

$$g(\lambda|\alpha, p) = \frac{e^{-\lambda/\alpha}\lambda^{p-1}}{\Gamma(p)\alpha^p}, \qquad 0 \leq \lambda < \infty,$$

where $\Gamma(p)$ is the gamma function of p. If p is a positive integer, $\Gamma(p) = (p-1)!$. For this distribution we obtain

$$E[\lambda] = p\alpha$$

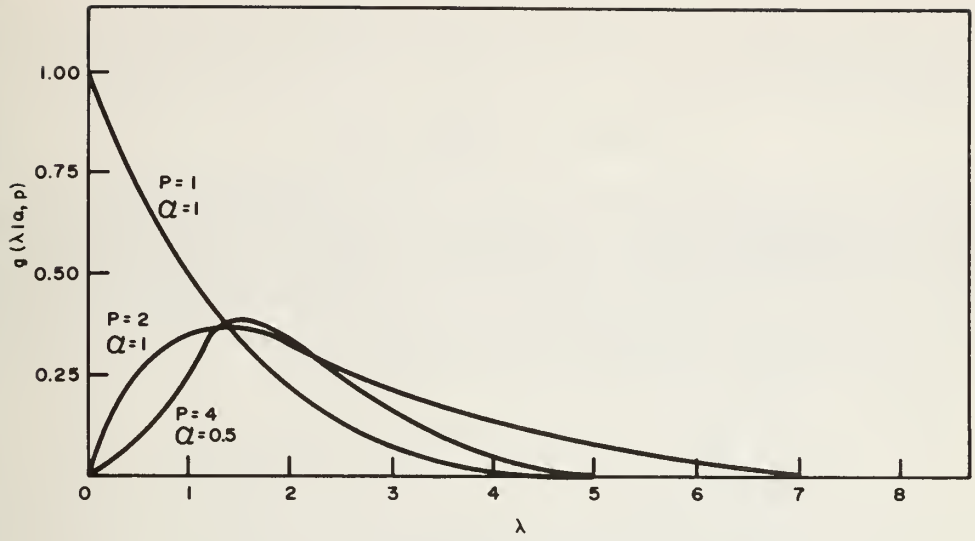
$$\mathrm{Var}[\lambda] = p\alpha^2,$$

and

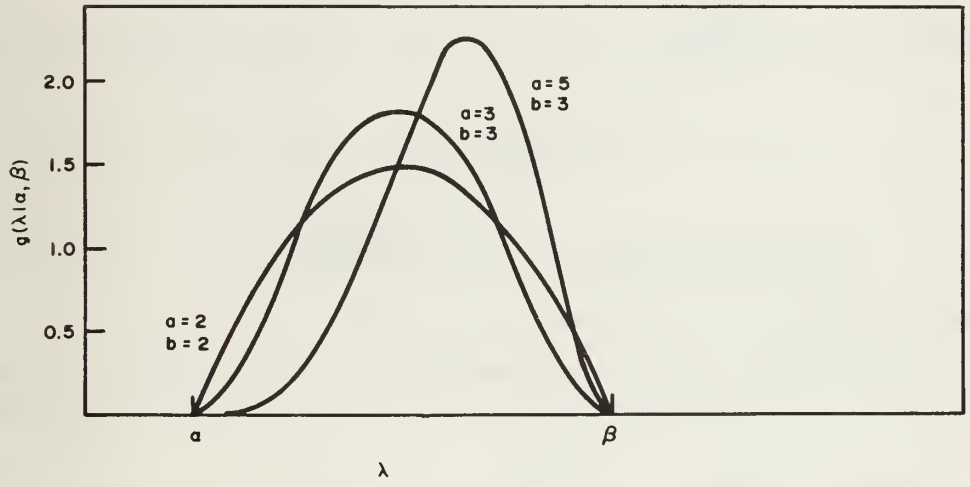
$$E[e^{-\lambda T}] = \left(\frac{1}{1+\alpha T}\right)^p.$$

The parameters (α, p) can easily be fit to approximate an empirical distribution by simply computing the mean and variance of the empirical distribution, μ and σ^2, and then solving for α and p to obtain

$$\alpha = \sigma^2/\mu, \qquad p = \mu^2/\sigma^2.$$



(a) Three gamma distributions.



(b) Three beta distributions.

FIGURE 1. Gamma and beta a priori density function.

The estimates are not maximum likelihood estimates for α and p, however. For numerical methods to obtain the maximum likelihood estimates, see Choi and Wette [1].

2.2.2 The Beta Distribution

The beta distribution is another possible candidate for the a priori distribution. It has also been used extensively in practical problems, see Good [4]. In the current context, however, it has the disadvantage of being less amenable to the mathematical manipulations than the gamma distribution.

The beta density is:

$$g(x|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} \qquad (0 \leq x \leq 1)$$

with

$$E[x] = \frac{a}{a+b}$$

$$\text{Var}[x] = \frac{ab}{(a+b)^2(a+b+1)}.$$

For application to the present problem we would not wish to restrict the replacement factor to the interval [0, 1] and would assume that $\lambda = \alpha + \beta x$ yielding a range for λ of $\alpha \leq \lambda \leq \beta$. This has the disadvantage of introducing two additional parameters α and β. Examples of some beta distributions are given in Figure 1b. For the beta distribution the characteristic function is not a simple function. However, $E[e^{-xT}]$ can be approximated by using the following identities:

$$E[e^{-xT}] = \sum_{0}^{\infty} \frac{E[x^k](-T)^k}{k!}$$

and

$$E[x^k] = \frac{\Gamma(a+b)\Gamma(a+k)}{\Gamma(a+b+k)\Gamma(a)}.$$

The convergence of the partial sum is as rapid as $\sum^{n} (-T)^k/k!$. To see this, note that for integer values of a and b

$$E[x^k] = \prod_{i=0}^{k-1} \frac{a+i}{a+b+i}$$

thus

$$E[x^k] < 1 \text{ and } E[x^k] \text{ strictly decreases as } k \text{ increases.}$$

2.3 Behavior of $E[B]$ and Var $[B]$ with Bayes Estimate of λ

The assumptions we make concerning the demand rate, λ, cause distinct differences in the mean and variance of the number of backorders. Using a point estimate for λ based on the available data

TABLE 1. *Comparison of Point to Bayes Estimate of Replacement Factor*

Stock policy (s)	Point estimate		Bayes estimate	
	$E[B]$	Var $[B]$	$E[B]$	Var $[B]$
0	$T\lambda$	$T\lambda$	$TE[\lambda]$	$T^2 \text{Var}[\lambda] + TE[\lambda]$
1	$T\lambda - (1 - e^{-T\lambda})$	$T - e^{-\lambda T}(2\lambda T + e^{-\lambda T} - 1)$	$TE[\lambda] - (1 - E[e^{-T\lambda}])$	$T^2 \text{Var}[\lambda] + TE[\lambda] - E[e^{-T\lambda}](2TE[\lambda] + E[e^{-T\lambda}] - 1)$
2	$T\lambda - (1 - e^{-T\lambda}) - (1 - e^{-T\lambda} - T\lambda e^{-T\lambda})$	$T\lambda - e^{-T\lambda}(2(T\lambda)^2 + T\lambda - 4) - e^{-2T\lambda}(T\lambda + 2)^2$	$TE[\lambda] - (1 - E[e^{-T\lambda}]) - (1 - E[e^{-T\lambda}] - E[T\lambda e^{-T\lambda}])$	$T^2 \text{Var}[\lambda] + E[T\lambda] + 4E[e^{-T\lambda}](1 - E[T\lambda] - E[e^{-T\lambda}] - E[T\lambda e^{-T\lambda}]) + E[T\lambda e^{-T\lambda}](3 - 2E[T\lambda] - E[T\lambda e^{-T\lambda}])$

T = Expected time to replenish stock

may lead to a different decision than if a Bayes estimate for λ is used, although it is also based on the same data. This is caused by the differences in $E[B]$ and Var $[B]$ which result from using the two different types of estimates of the replacement rate factor λ. The provisioning decision should of course make maximum use of all the data available. Using the Bayes, rather than the point estimate, will achieve this.

We have summarized $E[B]$ and Var $[B]$ in Table 1. This gives a good comparison for a general Bayes estimator of λ. The calculations are found in section 6. We also include Figure 2 to give a graphical comparison. For the Bayes estimates in this figure we have chosen a gamma distribution

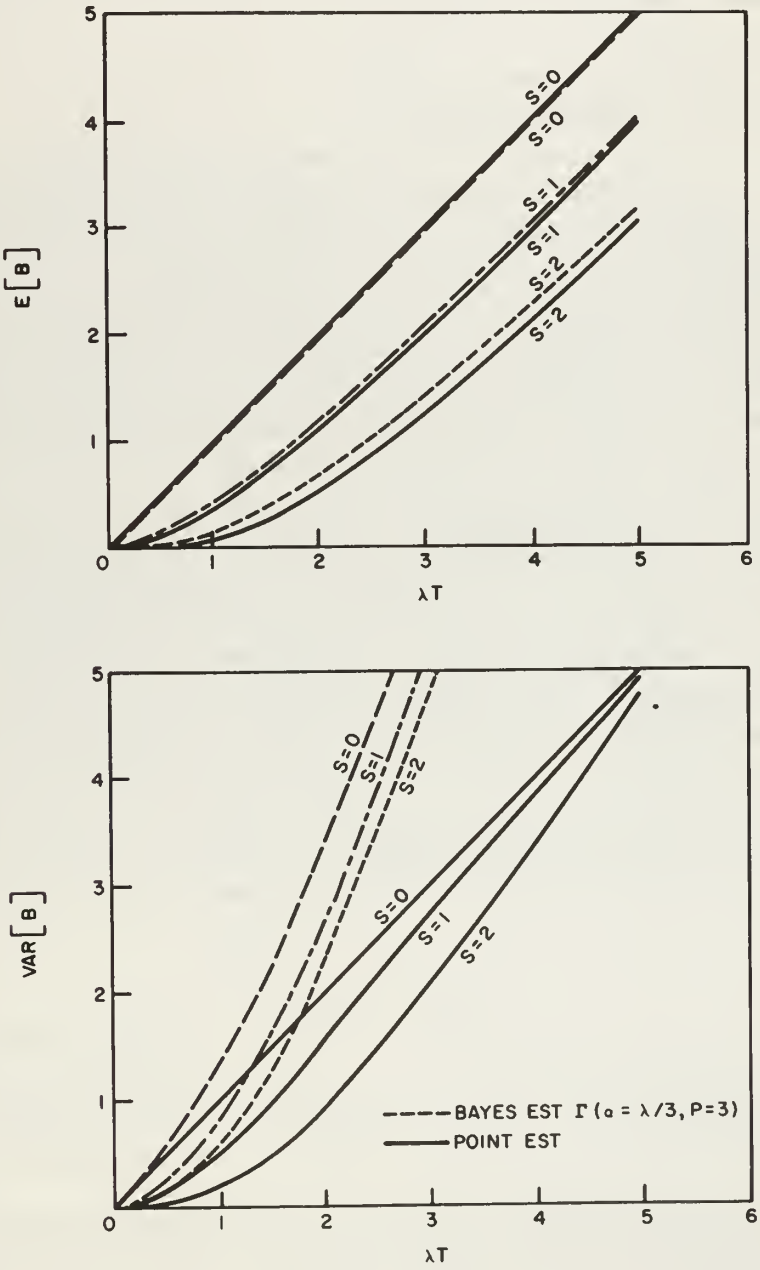


FIGURE 2. Comparison of $E[B]$ and Var $[B]$ for various stock levels.

with parameters of p equal to 3 and α chosen so that $E[\lambda]$ of the Bayes estimate is the same value as the point estimate. We have set $T=1$ arbitrarily.

3.0 SYSTEM OBJECTIVE

The decision to stock or not to stock an item must be based on some objective or criterion based on the overall supply system. One of the goals of this study is to develop several alternative criteria. We discuss four such criteria in this section. Each arrives at provisioning decisions by minimizing an additive criterion function within a budget constraint. We have given the following names to the functions considered:

1—Backorder Generation
2—Backorder Time
3—Time Weighted Backorders
4—Strongly Time Weighted Backorders.

Each of these criteria assigns a different utility or penalty to backorders, that is, each gives different relative penalties to such events as being short three items for 1 week, one short for 3 weeks, or one short for each of 3 separate weeks. Each of these possibilities has the same total number of weeks short, but they may each result in differing degrees of system degradation.

The system criteria finally selected should reflect the user's best estimate of how the system is degraded in the presence of shortages. In examining these four criteria, the reader should consider the relative penalties and determine which appropriately reflects the real system degradation. If, for example, the system can operate for a short time at full capacity with a shortage, but then degrades badly, then one short for 3 weeks should clearly have the greatest penalty. If the system can operate at full capacity with one short, but degrades badly with more than one short, then three short for 1 week should incur the greatest penalty. Evaluation of the effects of such events should be made without regard to any implied replacement rate and without regard to the likelihood of each of these events: these considerations are taken into account when expected values of the penalties are taken.

In sections 3.1 to 3.4 we discuss four criteria, and to understand the differences between the various decision criteria, it is useful to construct a penalty function for each criterion we consider. This function shows the instantaneous effects on the decision criterion for any given back order history. If we let B_i represent the actual backorders at any arbitrary point in time for item i, then the penalty that results, denoted P_i, depends on B_i, and of course on the particular criterion employed. B_i depends on many factors, including the resupply time, the stock level, and the replacement factor, however we supress this dependence in the notation. The initial provisioning problem can then be posed in the following form:

$$\text{minimize } \sum E[P_i]$$

subject to

$$\sum s_i c_i \leq C,$$

where s_i is the stock for the ith item, c_i is the cost of the ith item, and C is the total system budget. Note, if we were to include essentiality we would write $\underline{E}_i E[P_i]$ instead of $E[P_i]$ where $\underline{E}_i$ is the relative essentiality of item i.

3.1 Backorder Generation Criterion

This is a criterion that has been used in the past by certain portions of the Navy Supply System. The object of this criterion is to minimize the expected number of new backorders generated per unit time within the budget constraint. Under this criterion no account is taken of the time the backorder exists after being generated, and its use implies that each backorder degrades the system an equal amount whether it lasts for an instant or for a year.

It is somewhat difficult to write a penalty function for this first criterion in the same manner as for the other criteria considered below. At an arbitrary point in time we can represent the penalty, P_i, for the ith item as

$$P_i = \left(\frac{dB_i}{dt}\right)^+,$$

where $(x)^+ = x$ if $x > 0$ and $(x)^+ = 0$ if $x \leq 0$.

Figure 3 is a pictorial representation of this criterion's penalty function and is given here in discrete time for a particular backorder history. For this example $E[B_i] = 0.6$ in each 5-week period and for the 15-week period as well. For $E[P_i]$, however, we have values 0.6, 0.2, and 0.6 in the three 5-week periods and 0.47 for the entire 15-week period. For these expectations we have taken the average over time to represent the expectation at a point in time. These results derive from the primary characteristic of the criterion: each backorder counts only when it is created.

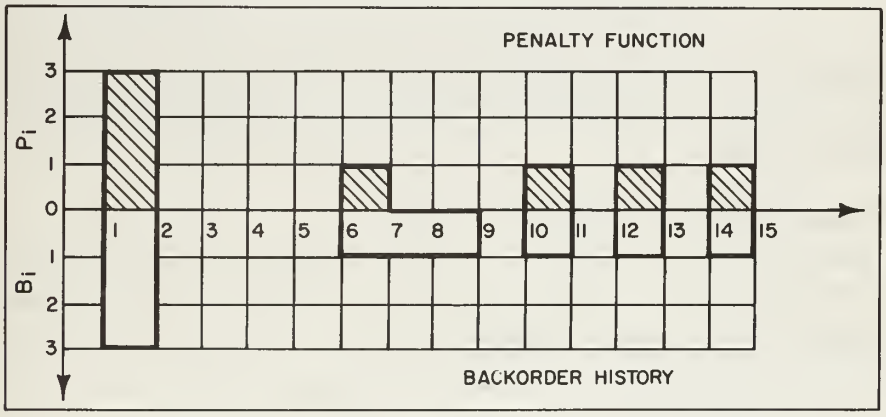


FIGURE 3. Backorder generation.

3.2 Backorder Time Criterion

This criterion penalizes the system when one or more backorders exist against an item. When a backorder exists, further demands for that item have no impact. To construct the penalty function we assign a penalty of 1 at each point in time if there are backorders and no penalty to a no backorder situation:

$$P_i = \begin{cases} 1 & B_i \geq 1 \\ 0 & B_i = 0. \end{cases}$$

For this criterion we have

$$E[P_i] = Pr\{B_i > 0\}$$

and

$$\mathrm{Var}[P_i] = [Pr\{B_i > 0\}] \cdot [Pr\{B_i = 0\}].$$

The pictorial representation of this penalty function for the same backorder history considered in section 3.1 is shown in Figure 4.

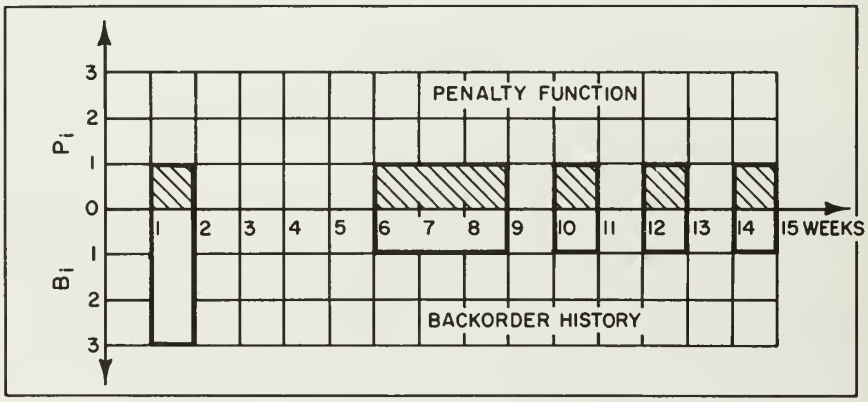


FIGURE 4. Backorder time.

For this example it is easy to see that $E[B_i] = 0.6$. Also the overall $E[P_i]$ is 0.47; it is 0.2 in the first 5 weeks, 0.6 in the second 5 weeks and 0.6 in the third. For this criterion three short for 1 week is equivalent to one short for 1 week.

3.3 Time Weighted Backorders Criterion

This is a standard system criterion for provisioning. For this criterion the penalty function P_i is just equal to B_i. Thus at any point in time, we assign a penalty weight of 1 to each backordered unit. Thus at any point in time the penalty for item i is

$$P_i = \begin{cases} \sum_{j=1}^{B_i} 1 = B_i & \text{if } B_i \geq 1 \\ 0 & \text{if } B_i = 0. \end{cases}$$

Observe that for low demand rate items this is very close to the penalty function associated with the backorder time criterion (Section 3.2). This is so since the probability that more than one backorder occurs, which is the probability that the penalty functions differ, is very small. This criterion can be compared with the previous two by examining Figure 5.

For this example $E[P_i] = E[B_i] = 0.6$ over the 15-week period as well as over each of the 5-week periods. This illustrates the characteristic of this criterion: one short for 3 weeks is equivalent to three short for 1 week.

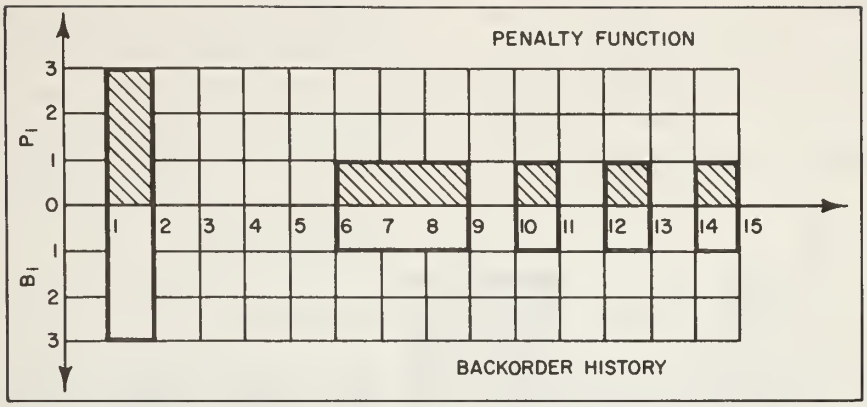


FIGURE 5. Time-weighted backorders

3.4 Strongly Time-Weighted Backorders Criterion

Time-weighted units short is a departure from many of the other formulations of the provisioning problem which only consider units short. In this strongly time-weighted framework, units which are short for a long period of time carry a heavier penalty than units short for a lesser period. Resupply time for equipment thus becomes a much more important factor in provisioning than would be true in the previous three criteria. Consider for some randomly chosen point in time the following variables:

t_{ij} = resupply time for the jth backorder on the ith equipment type.
B_i = number of backorders outstanding on the ith equipment type.

At that instant of time the penalty associated with the backorders is

$$P_i = \begin{cases} \sum_{j=1}^{B_i} t_{ij} & B_i \geq 1 \\ 0 & B_i = 0, \end{cases}$$

which represents the total backorder time for that equipment at that instant of time. For this criteria we have assigned a penalty weight of the backorder time to each backorder.

For this criterion we can make the simplifying assumption that $E[t_{ij}] = T_i$ independent of the number of backorders, B_i, or which backorder, j. (This assumption is probably not justified when the number of units in backorder is large, when one might expect the resupply time to increase. For the stock-no stock problem, however, this assumption should be of little consequence since the number in backorder at any time will always be small.) If we make this assumption, we can write $E[P_i]$ as

$$E[P_i] = E[B_i]T_i$$

and find

$$\text{Var } [P_i] = E[B_i] \text{ Var } [t_i] + T_i^2 \text{ Var } [B_i).$$

We can compare this criterion with the previous one by examining Figure 6, where we assume $t_{ij} = T_i$.

Again $E[B_i] = 0.6$ in the 15-week period and in each of the 5-week periods. However, the overall time-weighted penalty factor is $E[P_i] = 1.0$. It has the value $E[P_i] = 0.6$ for the first and last 5-week period, and $E[P_i] = 1.8$ in the second 5 weeks. Thus one short for 3 weeks is three times as bad as three short for 1 week, or three times as bad as one short for 3 disjoint weeks.

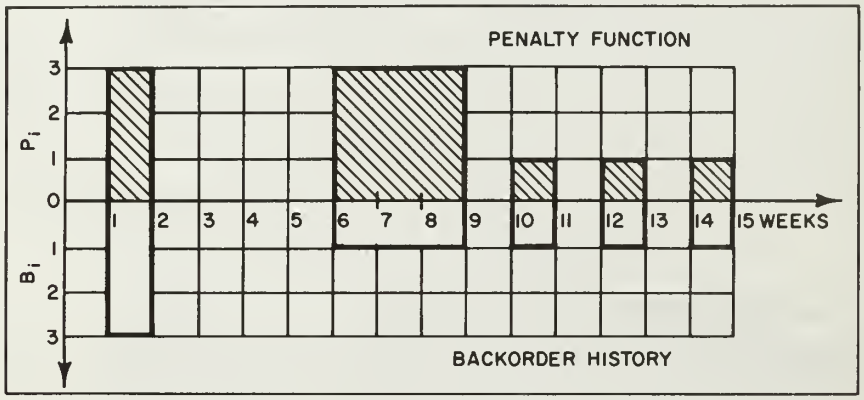


FIGURE 6. Strongly time-weighted backorders.

4.0 INITIAL PROVISIONING WITHIN A FIXED BUDGET

The problem which we discuss in this section is how to allocate the budget: to decide which items to stock and which not to stock within a fixed budgetary limit. The decision will be based on some overall system objective, as discussed in section 3. This problem is a problem in resource allocation and can be formulated mathematically in the form

$$\text{minimize } \sum f_i(s_i) \qquad \text{(the system criterion or penalty)}$$

(4.1)

$$\text{subject to} \qquad \sum c_i s_i \leq C \qquad \text{(the budget constraint),}$$

where

- $f_i(\cdot)$ is the additive criterion (penalty) function for item i
- s_i is the number of units of i purchased and stocked
- c_i is the unit price of item i
- C is the total budget.

In this formulation we have not yet restricted s_i to be 0 or 1.

The problem of the existence of an optimal solution to this problem (4.1) has been studied extensively, e.g., Everett [2]. In Everett's paper Lagrange multiplier methods are discussed and some results are proved on the existence and construction of optimal solutions. Rather than give a detailed review of that material, we can paraphrase these results for our particular problem. If there exists a Lagrangian γ^* and a set of s_i^* such that these s_i^* minimize the functions $f_i(s_i) + \gamma^* \Sigma c_j s_j$ for all i, then there is no other selection of the s_i satisfying $\Sigma c_i s_i \leq \Sigma c_i s_i^*$ and $\Sigma f_i(s_i) \leq \Sigma f_i(s_i^*)$. If we assume that the $f_i(s_i)$ are convex,

which is not particularly restrictive for reasonable criterion, we can guarantee the existence of suitable γ^* and can state this result more explicitly. Define

$$\gamma_i(s_i) \equiv [f_i(s_i) - f_i(s_i+1)]/c_i,$$

define $s_i(\gamma)$ to be the s_i satisfying

$$\gamma_i(s_i) \leq \gamma \leq \gamma_i(s_i-1),$$

and define

$$C(\gamma) \equiv \sum c_i s_i(\gamma).$$

Then we know that no selection of s_i other than $s_i(\gamma)$ can satisfy $\Sigma c_i s_i \leq C(\gamma)$ and $\Sigma f_i(s_i) \leq \Sigma f_i[s_i(\gamma)]$. Thus if we can find a γ^* such that $C(\gamma^*) = C$, then $s_i^* = s_i(\gamma^*)$ is optimal.

These results do not guarantee an easy method for obtaining the optimal s_i^* solving the initial allocation problem. The problem arises if $C(\gamma) \leq C < C(\gamma - \delta)$ for all $\delta > 0$. Methods for solving the allocation problem (4.1) exactly do exist. These refinements, such as the "Noose Method" discussed Furman [3], are somewhat complex. In the context of a large supply system, the relatively small portion of the budget which cannot be allocated optimally by a straight-forward application of techniques suggested by Everett's results does not warrant the considerable increase in effort.

The resource allocation which we shall use to solve (4.1) may be suboptimal. The procedure which provides this allocation has two parts; the first is suggested by Everett's results and the second deals with the unallocated resources after applying the first part:

1. Find a γ^* such that $C(\gamma^*) \leq C < C(\gamma^* - \delta)$ for all $\delta > 0$. The $s_i(\gamma^*)$ would be optimal for the smaller budget $C(\gamma^*)$.

2. Allocate the remaining $C - C(\gamma^*)$ as follows: Lower γ below γ^*; this will increase the $s_i(\gamma)$ above $s_i(\gamma^*)$ in discrete jumps of a unit each, and we know that the first $s_i(\gamma)$ thus increased cannot be bought within the remaining funds. We can lower γ, however, and ignore all incremental purchases that cannot be afforded with the funds then remaining, but increase the indicated purchase of the item if funds are available.

If the budget allocation problem is restricted so that we only allow s_i to be zero or one, the formulation and solution becomes very straight-forward. Simply let

$$\gamma_i = [f_i(0) - f_i(1)]/c_i,$$

and rank the items in order of decreasing γ_i: let $\gamma_{i_1} \geq \gamma_{i_2} \geq \cdots \geq \gamma_{i_N}$. Then find the index k such that

$$\sum_{j=1}^{k} c_{i_j} \leq C < \sum_{j=1}^{k+1} c_{i_j}.$$

The initial solution, Part 1 of the above procedure, is to stock (buy one each of) items $i_1, \ldots, i_k$ and not to stock (buy none of) items $i_{k+1}, \ldots, i_N$.

In general a portion of the budget

$$C - \sum_{j=1}^{k} c_{i_j}.$$

will not be allocated by this method. In most cases this amount can be used to improve the overall system performance according to Part 2 of the above procedure. Simply continue along the list through items $i_{k+1}, i_{k+2}, \ldots, i_N$ and when considering each item in turn, if its unit price exceeds the remaining unallocated funds, skip the item and move to the next one. If funds are available, add the item to the buy list and reduce the unallocated funds by the item price. The procedure is simply to buy in order from largest to smallest γ until the money runs out.

If the functions $f_i(s)$ are "well behaved," then

$$f_i(s-1) - f_i(s) \geq f_i(s) - f_i(s+1).$$

When this is so, we can easily generalize the zero-one problem (stock-no stock) above to a zero-one-two problem. Simply double the number of apparent items and define

$$\gamma_i = [f_i(0) - f_i(1)]/c_i \qquad i = 1, \ldots, N$$

$$\gamma_i = [f_{i-N}(1) - f_{i-N}(2)]/c_{i-N} \qquad i = N+1, \ldots, 2N.$$

Treating this as a zero-one problem gives us the optimal solution, where buying item $N+j$ means buying the second unit of item j. We are guaranteed by assumption that if the second unit of an item is bought, the first has already been bought, i.e., $\gamma_j \geq \gamma_{N+j}$.

In the following sections we shall formulate the allocation procedure for each of the four criterion functions discussed in section 3.

4.1 Expected Backorder Generation

Using this criterion of minimizing the total system expected backorders generated per unit time (see section 3.1), we have

$$\text{minimize} \quad \sum E[BG_i]$$

subject to

$$\sum s_i c_i \leq C,$$

where BG_i is the penalty function given in section 3.1. We assume that the replacement rate, λ_i, is given by a gamma a priori distribution characterized by the parameters (α_i, p_i). To calculate the Lagrange multipliers we first determine the form of the penalty function:

$$\left(\frac{dB_i}{dt}\right)^+ = Pr\{\text{there is zero stock}\} \cdot E[\text{demands in time } \tau]/\tau \qquad \text{for small } \tau$$

$$= \left(\sum_{k=s}^{\infty} \frac{(\lambda T)^k e^{-\lambda T}}{k!}\right)\left(\sum_{k=1}^{\infty} \frac{k(\lambda\tau)^k e^{-\lambda\tau}}{\tau k!}\right) \qquad \text{for small } \tau$$

$$= \lambda \sum_{k=s}^{\infty} \frac{(\lambda T)^k e^{-\lambda T}}{k!},$$

therefore

$$\gamma_i = \{E[BG_i(0)] - E[BG_i(1)]\}/c_i$$

$$= E[\lambda_i \exp(-\lambda_i T_i)]/c_i$$

$$= \alpha_i p_i \left(\frac{1}{1+\alpha_i T_i}\right)^{p_i+1}/c_i.$$

4.2 Expected Backorder Time

Using this criterion of minimizing the total system expected backorder time (see section 3.2), we have

$$\text{minimize} \sum E(BT_i)$$

subject to

$$\sum s_i c_i \leq C,$$

where BT_i is the penalty function given in section 3.2. Holding to the same assumptions as above we can calculate the Lagrange multipliers for this criterion:

$$\gamma_i = \{E[BT_i(0)] - E[BT_i(1)]\}/c_i$$

$$= [Pr\{B_i(0) > 0\} - Pr\{B_i(1) > 0\}]/c_i$$

$$= E[\lambda_i T_i \exp(-\lambda_i T_i)]/c_i$$

$$= \alpha_i p_i T_i \left(\frac{1}{1+\alpha_i T_i}\right)^{p_i+1}/c_i.$$

4.3 Expected Time Weighted Backorders

Using the criterion of minimizing the total system expected backorders (see section 3.3), we have

$$\text{minimize} \sum E[B_i]$$

subject to

$$\sum c_i s_i \leq C.$$

We calculate the Lagrange multipliers in this case as follows:

$$\gamma_i = \{E[B_i(0)] - E[B_i(1)]\}/c_i$$

$$= \{1 - E[\exp(-\lambda_i T_i)]\}/c_i$$

$$= \left[1 - \left(\frac{1}{1+\alpha_i T_i}\right)^{p_i}\right]/c_i,$$

where T_i is the expected resupply time for the ith item.

4.4 Expected Strongly Time-Weighted Backorders

Using the criterion of minimizing the total system expected time-weighted backorders (see section 3.4), we have

$$\text{minimize} \sum E[TB_i]$$

subject to

$$\sum c_i s_i \leq C,$$

where TB_i is the penalty function of section 3.4. Again assuming that the backorder times, T_i, are not a function of the number of backorders and also that λ_i is given by a gamma distribution with parameters (α_i, p_i), we can calculate the Lagrange multipliers to use in the procedure of section 4.0 to determine stocking requirements. For this case:

$$\gamma_i = \{E[TB_i(0)] - E[TB_i(1)]\}/c_i$$

$$= \{1 - E[\exp(-\lambda_i T_i)]\}T_i/c_i$$

$$= \left[1 - \left(\frac{1}{1+\alpha_i T_i}\right)^{p_i}\right] T_i/c_i.$$

4.5 Behavior and Comparison of the Criterion Functions

The Lagrangian value, γ, computed for each of the items in the insurance item category, determines its stocking desirability. The higher the value the more desirable it is to stock. The behavior of the Lagrangian as a function of its provisioning parameter is presented in Figure 7 for each of the four criterion functions discussed above. These plots show for each of the four criteria how the demand estimate and the expected resupply time, T, affect the Lagrangian. Each plot also shows how "uncertainty" in the demand estimate affects the Lagrangian. Uncertainty has been measured as the "coefficient of variation" (standard deviation/mean) of the a priori demand distribution. For the gamma distribution this is $1/\sqrt{p}$. A large coefficient of variation suggests a wide variation between demand rate estimates and thus a great deal of uncertainty in knowing its actual value. For these comparisons, item cost has been set to unity.

The plots in Figure 7 illustrate the following interesting characteristics of the various criterion functions:

(1) Backorder Generation (section 4.1 and top three plots)

- Reorder time has a large effect on this criterion function. The direction of this effect, however, is the reverse of what one would intuitively like; i.e., as the reorder time decreases the Lagrangian (desirability for stocking) increases.
- Uncertainty about the demand rate has a larger effect when the reorder time is short than when it is long.
- When the reorder time and coefficient of variation are held fixed, the criterion function reaches a maximum when Demand = 1/Reorder Time.

(2) Backorder Time (section 4.2 and 2nd row of plots)

● Reorder time does not have much impact on the decision to stock or not to stock the item.

● Like "Backorder Generation," when the reorder time and coefficient of variation are fixed, the criterion function reaches a maximum when the demand rate is the reciprocal of the reorder time.

(3) Time-Weighted Backorders (section 4.3 and 3rd row of plots)

● The stocking desirability of an item increases as the reorder time increases.

● The impact of the demand rate uncertainty factor also increases as the reorder time increases.

(4) Strongly Time-Weighted Backorders (section 4.4 and last row of plots)

● The characteristics of these curves are similar to "Time-Weighted Backorders," only the reorder time has a much greater impact.

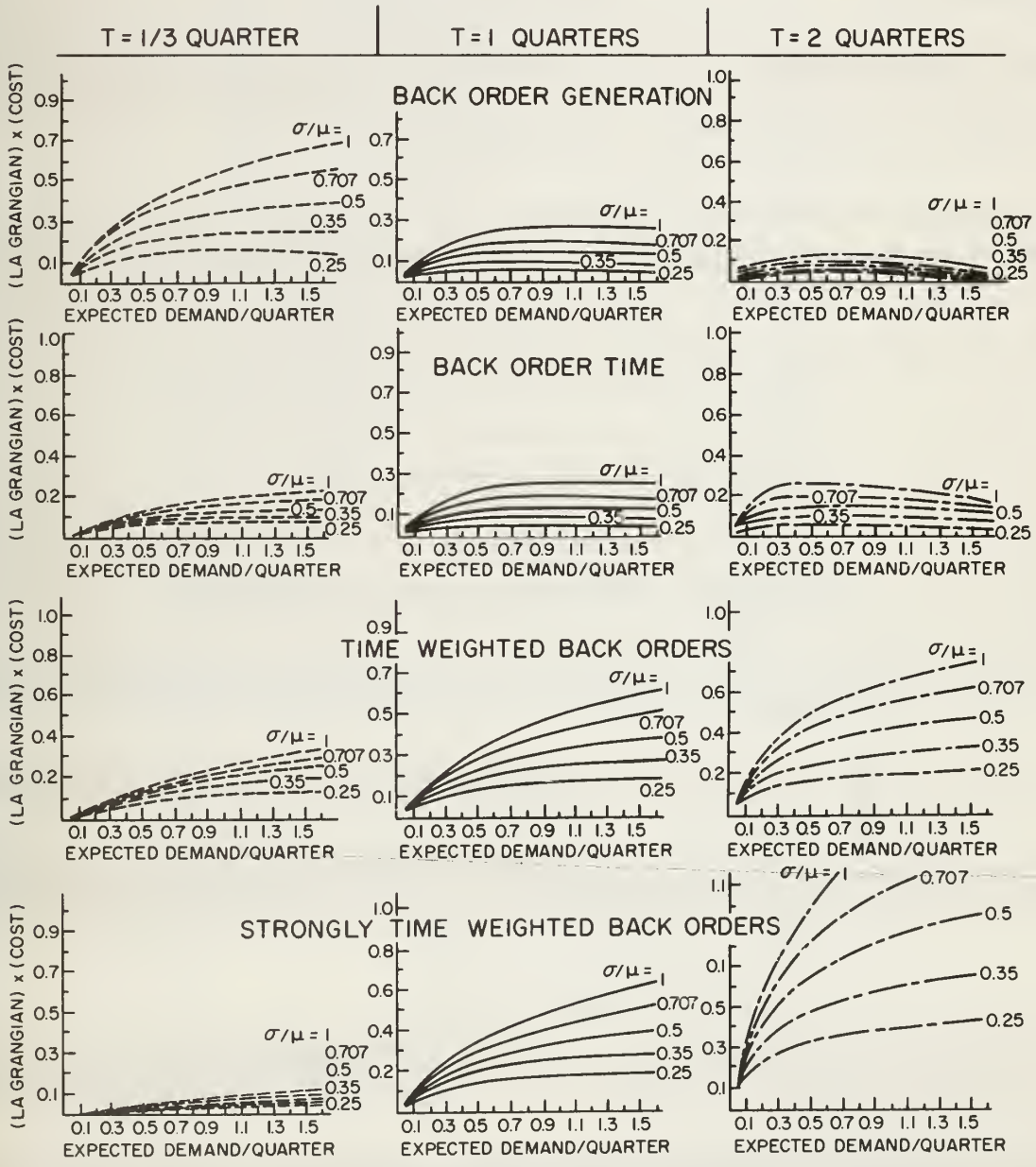


FIGURE 7. Comparison of criteria.

4.6 Time Dependence Comparisons

We can also compare these four criteria on the basis of how heavily they weight the procurement lead time. And, in fact, we can rank the four criteria on their sensitivity to lead time. Using this ranking the decision maker can select the criterion which best reflects his knowledge of how the system is degraded with the time to fill backorders.

The four criteria and their γ_i values are:

1. Backorder Generation $\alpha_i p_i \left(\frac{1}{1+\alpha_i T_i}\right)^{p_i+1} / c_i;$
2. Backorder Time $\alpha_i p_i T_i \left(\frac{1}{1+\alpha_i T_i}\right)^{p_i+1} / c_i;$
3. Time-Weighted Backorders $\left[1-\left(\frac{1}{1+\alpha_i T_i}\right)^{p_i}\right] / c_i;$
4. Strongly Time-Weighted Backorders $\left[1-\left(\frac{1}{1+\alpha_i T_i}\right)^{p_i}\right] T_i / c_i.$

Since dollars are allocated to items with large γ_i values, the γ_i give a reasonable comparative measure on the propensity to stock an item. Figures 8 and 9 give comparisons to show the sensitivity

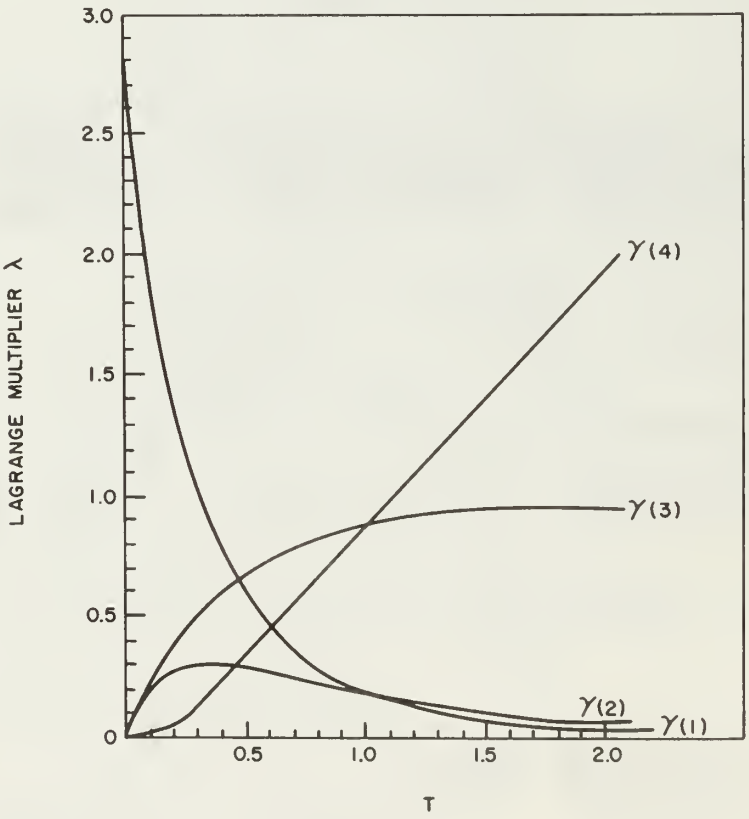


FIGURE 8. Lagrange multiplier for various criteria.

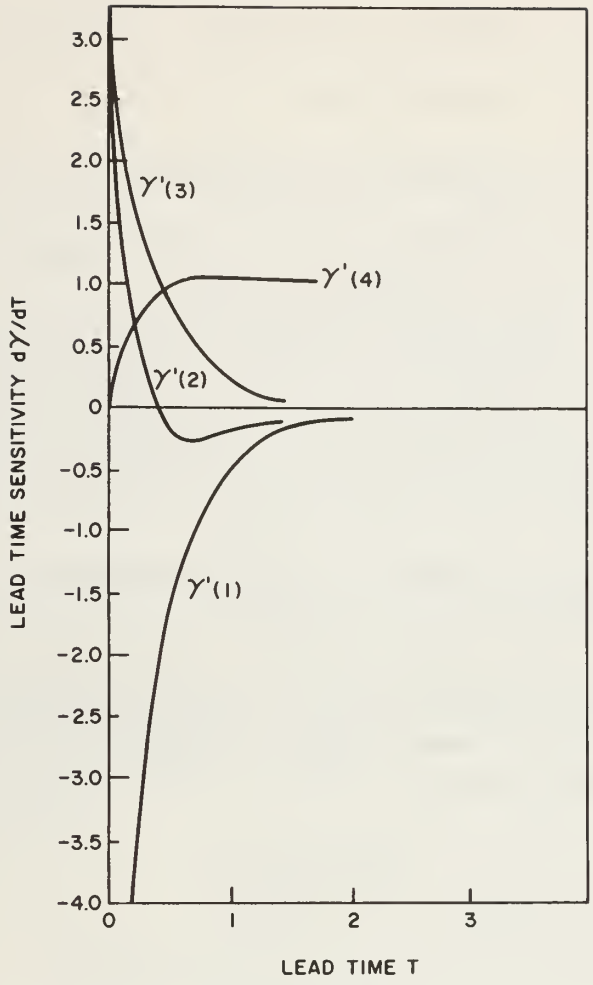


FIGURE 9. Lead time sensitivity for Lagrange multiplier for various criteria.

of γ to the lead time T. We have taken c and α to be unity, which we can do without loss of generality. We have arbitrarily taken p to be 3. The stock-no stock test functions, γ, then become

$$\gamma(1) = 3(1+T)^{-5}$$

$$\gamma(2) = 3T(1+T)^{-4}$$

$$\gamma(3) = 1-(1+T)^{-3}$$

$$\gamma(4) = T[1-(1+T)^{-3}],$$

and their marginal sensitivity to T, $\frac{d\gamma}{dT}$, become

$$\gamma'(1) = -12(1+T)^{-5}$$

$$\gamma'(2) = 3(1-3T)(1+T)^{-5}$$

$$\gamma'(3) = 3(1+T)^{-4}$$

$$\gamma'(4) = 1+(2T-1)(1+T)^{-4}.$$

Figure 8 shows the γ functions, the propensity to stock an item for various T, and Figure 9 shows the γ' functions, the marginal propensity to stock an item as T varies.

The significant features of these graphs and functions are that $\gamma(2) = T\gamma(1)$ and $\gamma(4) = T\gamma(3)$; that $\gamma(3)$ and $\gamma(4)$ are increasing in T while $\gamma(1)$ is decreasing in T; that $\gamma'(3)$ and $\gamma'(1)$ are asymptotic monotonic increasing functions whereas $\gamma'(4)$ is asymptotically monotonically decreasing; and that $\gamma(2)$ and $\gamma'(2)$ have nonmonotonic behavior. From these observations we can rank these four criteria by dependence on T. From strongest tendency to stock long lead time items to least tendency, the criteria are ranked 4, 3, 2, 1.

5.0 INITIAL PROVISIONING WITHIN A SYSTEM PERFORMANCE CONSTRAINT

In section 4 we discussed the provisioning problem in terms of a resource allocation problem where the objective is to obtain maximum system performance within the fixed budget available. In this section, we discuss the dual mathematical problem to solve the provisioning problem. The object is to expend a minimum amount of money to meet a fixed minimum system performance level. The mathematics of the problem are again quite simple.

$$\text{Minimize } \sum c_i s_i$$

(5.1)

$$\text{subject to} \qquad \sum f_i(s_i) \geq L,$$

where the new variable, L, is the minimum acceptable level of system performance.

For the stock-no stock problem Equation (5.1) reduces to a simple form similar to that discussed in section 4. Define

$$v_i = c_i/[f_i(0) - f_i(1)],$$

and rank the items in order of increasing v_i: $v_{i_1} \leq v_{i_2} \leq \ldots \leq v_{i_N}$. Then find the index k such that

$$\sum_{j=1}^{k} f_{i_j}(1) + \sum_{j=k+1}^{N} f_{i_j}(0) \geq L > \sum_{j=1}^{k-1} f_{i_j}(1) + \sum_{j=k}^{N} f_{i_j}(0).$$

The stocking policy is to stock items i_1 through i_k and not to stock the remaining items.

Again, this is a suboptimal solution, however, the difference in cost between this solution and the "optimal" solution should not exceed c_{i_k}. The procedure is simply to stock items from lowest v_i to largest v_i until the system performance level is met. It should be clear that the order of stocking is the

same, i.e., the same indexing gives

$$v_{i_1} \leqslant v_{i_2} \leqslant \ldots \leqslant v_{i_N} \qquad \text{and } \gamma_{i_1} \geqslant \gamma_{i_2} \geqslant \ldots \geqslant \gamma_{i_N}.$$

6.0 DERIVATION OF EXPECTED BACKORDER EQUATIONS

Derivations of the equations presented in Table 1 are developed in this section. The main assumption used to derive these expected backorders is that the occurrence of demands in time can be represented by a Poisson process. For almost all low demand items this a priori assumption should be sufficiently accurate so that any errors caused by it should be minimal. In fact, in the time span of interest it would be difficult to discern the existence of other kinds of demand processes in operation.

For a Poisson demand process, it can be shown that the number of unfilled demands that exist at any point in time under steady state conditions is again a Poisson process. The probability distribution of the backorders is Poisson with parameter equal to the demand rate of the demand process times the average (or expected) time to resupply the item. The result that it is the expected resupply time, T, no matter what a priori distribution the true resupply time has, which enters into the backorder distribution parameter, is somewhat subtle, and can be found in Parzen [6, chapter 4].

Defining,

B = random variable denoting the number of backorders at an arbitrary point in time;

T = expected resupply (repair) time;

λ = mean demand per unit time, e.g. (1/BRF);

s = stocking level; and

then for a given stocking level, s,

$$P(B=b|\lambda) = \begin{cases} \sum_{k=0}^{s} \dfrac{(\lambda T)^k e^{-\lambda T}}{k!} & \text{for } b=0 \\ \dfrac{(\lambda T)^{s+b} e^{-\lambda T}}{(s+b)!} & \text{for } b \neq 0. \end{cases} \tag{6.1}$$

Associated with λ will be a probable range of occurrence. Let $g(\lambda)$ denote the probability density function for λ over this range.

If we use a point estimate for λ the expected number of backorders at any point in time is represented by

$$E(B) = E(B|\lambda) \qquad \text{[for point estimate]}.$$

For the case where a distribution estimate is used for λ (Bayesian) then:

$$E(B) = E[E(B|\lambda)] \qquad \text{[Bayesian Estimate]}.$$

Computation of $E(B|\lambda)$ and Var $(B|\lambda)$ for Stock Levels of Interest in the Stock-No Stock Problem

CASE 0: $s=0$

For this case B_λ (B given the value λ) has a Poisson distribution, thus

$$E(B|\lambda) = T$$

and

$$\text{Var}\,(B|\lambda) = T\lambda.$$

Using the $g(\lambda)$ distribution,

$$E(B) = E(E(B|\lambda)) = E(T\lambda) = TE(\lambda)$$

and

$$\text{Var}\,(B) = TE\lambda + T^2\,\text{Var}\,\lambda.$$

This last equality can be developed as follows:

$$\text{Var}\,(B) = EB^2 - (EB)^2,$$

but

$$EB^2 = E[EB^2|\lambda] = E\{T\lambda + (T\lambda)^2\}$$

$$= TE\lambda + T^2E\lambda^2.$$

Therefore,

$$\text{Var}\,B = TE\lambda + T^2E\lambda^2 - (TE\lambda)^2,$$

which yields the above result.

CASE 1. $s=1$

The use of Equation (1) with $s=1$ yields the following results:

$$E(B+1|\lambda) = 1 \cdot \sum_0^1 \frac{(\lambda T)^k e^{-\lambda T}}{k!} + \sum_1^\infty \frac{(b+1)(\lambda T)^{b+1}e^{-\lambda T}}{(b+1)!} - e^{-\lambda T} + \lambda T e^{-\lambda T} + \lambda T[1-e^{-\lambda T}],$$

therefore,

$$E(B|\lambda) = \lambda T - [1-e^{-\lambda T}].$$

To compute the variance term we first look at

$$E[B(B+1)|\lambda] = \sum_1^\infty \frac{b(b+1)(\lambda T)^{b+1}e^{-\lambda T}}{(b+1)!} = (\lambda T)^2 .$$

$$\text{Var}\,(B|\lambda) = E(B^2|\lambda) - (EB|\lambda)^2 = E(B(B+1)|\lambda) - E(B|\lambda) - (EB|\lambda)^2$$

$$= (\lambda T)^2 - [\lambda T - (1-e^{-\lambda T})][\lambda T + e^{-\lambda T}]$$

$$= \lambda T - e^{-\lambda T}[2\lambda T + e^{-\lambda T} - 1].$$

For the case where a distribution estimate is used for λ (Bayesian),

$$EB = E[E(B|\lambda)] = E\{T\lambda - [1-e^{-T\lambda}]\}$$

$$= TE\lambda - [1 - E(e^{-T\lambda})],$$

and the variance term can be computed from the relation

$$\operatorname{Var} B = E[B(B+1)] - EB - (EB)^2.$$

Since

$$E[B(B+1)] = E\{EB(B+1)|\lambda\} = E(T^2\lambda^2) = T^2E(\lambda^2)$$

we have

$$\operatorname{Var} B = T^2 \operatorname{Var} \lambda + TE\lambda - Ee^{-T\lambda}(2TE\lambda + E(e^{-T\lambda}).$$

The mean and variance expressions contain the term $E(e^{-T\lambda})$. Since this term is closely associated with the characteristic function of the distribution for λ, it is easily computed when the characteristic function is of simple form. For other cases, the value can be approximated using the moments of the distribution.

CASE 2: $s=2$

$$E(B+2|\lambda) = 2\sum_{k=0}^{2}\frac{(\lambda T)^k e^{-\lambda T}}{k!} + \sum_{b=1}^{\infty}\frac{(b+2)\lambda T^{b+2}e^{-\lambda T}}{(b+2)!}$$

$$= 2\left[e^{-\lambda T} + \lambda T e^{-\lambda T} + \frac{(\lambda T)^2 e^{-\lambda T}}{2!}\right] + \lambda T[1 - e^{-\lambda T} - \lambda T e^{-\lambda T}]$$

$$= \lambda T + 2e^{-\lambda T} + \lambda T e^{-\lambda T}.$$

Therefore,

$$E(B|\lambda) = T\lambda - (1 - e^{-T\lambda}) - (1 - e^{-T\lambda} - T\lambda e^{-T\lambda})$$

$$E\{(B+1)(B+2)|\lambda\} = \sum_{1}^{\infty}\frac{(b+1)(b+2)(\lambda T)^{b+2}e^{-\lambda T}}{(b+2)!} + 2\sum_{k=0}^{2}\frac{(\lambda T)^k e^{-\lambda T}}{k!}$$

$$= (\lambda T)^2 + 2[e^{-\lambda T} + \lambda T e^{-\lambda T}].$$

Therefore,

$$\operatorname{Var}(B|\lambda) = E\{(B+1)(B+2)|\lambda\} - E(B|\lambda)[3 + E(B|\lambda)] - 2$$

$$= (T\lambda)^2 + 2[e^{-T\lambda} + T\lambda e^{-T\lambda}] - 2 - (T\lambda + 2e^{-T\lambda} + T\lambda e^{-T\lambda} - 2)(T\lambda + 2e^{-T\lambda} + T\lambda e^{-\lambda} + 1)$$

$$= (T\lambda)^2 - (T\lambda + 2e^{-T\lambda} + T\lambda e^{-T\lambda})^2 + T\lambda + 4e^{-T\lambda} + eT\lambda e^{-T\lambda},$$

or

$$\operatorname{Var}(B|\lambda) = T\lambda - e^{-T\lambda}(2(T\lambda)^2 + T\lambda - 4) - e^{-2T\lambda}(T\lambda + 2)^2.$$

For the case where a distribution estimate is used for λ

$$E(B) = E(EB|\lambda) = TE\lambda - [1 - E(e^{-T\lambda})] - [1 - E(e^{-T\lambda}) - E(T\lambda e^{-T\lambda})].$$

For the variance term

$$\begin{aligned}
\mathrm{Var}(B) &= E[(B+1)(B+2)] - E(B)[3+E(B)] - 2 \\
&= E\{E[(B+1)(B+2)|\lambda]\} - E(B)[3+E(B)] - 2 \\
&= T^2 E\lambda^2 + 2[Ee^{-\lambda T} + E(\lambda T e^{-\lambda T})] - [E(T\lambda) + 2E[e^{-T\lambda}] + E[T\lambda e^{-T\lambda}] - 2]\,[E[T\lambda] \\
&\quad + 2E[e^{-T\lambda}] + E(T\lambda e^{-T\lambda}) + 1] - 2 \\
&= T^2\,\mathrm{Var}\lambda + TE\lambda + 4E(e^{-T\lambda})[1 - ET\lambda - Ee^{-T\lambda} - ET\lambda e^{-T\lambda}] \\
&\quad + E(T\lambda e^{-T\lambda})[3 - 2ET\lambda - ET\lambda e^{-T\lambda}].
\end{aligned}$$

Again, the values for $E(e^{-T\lambda})$ and $E(T\lambda e^{-T\lambda})$ can be easily computed if the characteristic function of λ is of simple form.

BIBLIOGRAPHY

[1] Choi, S. C. and R. Wette, "Maximum Likelihood Estimation of the Parameters of the Gamma Distribution and Their Bias," Technometrics, *11*, 683–690 (1969).

[2] Everett, H., III, "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources," Operations Research, *11*, 399–417 (1963).

[3] Furman, G. R., "Program OPSTRA," Stanford Research Institute Technical Report No. 5205–34 (1968).

[4] Good, I. J., *The Estimation of Probabilities* (Research Monograph No. 30), (M.I.T. Press, Cambridge, Massachusetts, 1965).

[5] Mood, A. M. and F. A. Graybill, *Introduction to the Theory of Statistics* (McGraw Hill, New York, 1963).

[6] Parzen, E., *Stochastic Processes* (Holden-Day, San Francisco, 1962).

[7] Sherbrooke, C. C., "Metric: A Multi-Echelon Technique for Recoverable Item Control," Operations Research *16*, 122–141 (1968).

DETERMINATION OF THE OPTIMAL INVESTMENT IN END PRODUCTS AND REPAIR RESOURCES*

Marshall Rose

Environmental Protection Agency

ABSTRACT

This paper demonstrates how to determine the minimal cost combination of end products and repair service capability in order to maintain a given level of operating end products. It is shown that the general rule for optimal resource allocation requires simply that the absolute value of the marginal factor cost of repair services divided by the marginal factor cost of end products be equal to the arrival rate of end products to repair. The model is applied to the problem of determining the resources required to support an operating level of Navy F–4 aircraft.

INTRODUCTION

Consider a military or commercial enterprise which maintains a fleet of operational end products and an internal repair facility for periodic service of the end products. Repair resources include items such as spares, equipment, and labor. Suppose the current number of operating end products falls below the required level. The problem is how to eliminate this deficiency.

The required operating inventory can be restored by 1) purchasing additional end products; 2) reducing the number of end products tied up in servicing and thus unavailable for operation; or 3) some combination of both. According to the first approach, the number of end products which must be acquired would have to exceed the number of end products added to the operating inventory, since some percent of the newly acquired end products would, on an average, be in repair. When the second approach was used, the reduction in the number of end products in repair could be obtained by reducing the service time per rework. The main objective of this paper is therefore to determine the optimal combination of end products and repair resources to maintain a specified number of operating end products at least cost per unit time. Two methods of achieving this objective are described in subsequent sections.

These methods employ well known mathematical tools. Hence, the main contribution of this paper is the presentation of a model, using fairly simple techniques, that can be applied in practice, to an important resource allocation problem. As such, this paper is of primary interest to the practicing operations research analyst.

Although the problem being discussed is well known, little guidance is provided in the literature for its solution, or for application of the solution to a real world problem. The concluding section of this paper applies the model to the case of the repair of Navy combat aircraft.

---

*The majority of the work needed to complete this paper was accomplished during the author's tenure at the Center for Naval Analyses.

OPTIMIZING BEHAVIOR OF THE FIRM

The following notation will be used:

A – the total number of (operating and nonoperating) end products;
K – the (deterministic) number of days needed to service the end products;
N – the number of operating end products;
H' – the fixed number of days for the end-product tour length;
$\Upsilon(A)$ – the (periodic) cost of having A end products; and,*
$\Phi(K, N)$ – the (periodic) cost of servicing end products in K days when the operating level is N.

We assume that the quantities to be optimized are continuous functions of the decision variables, A and K, with continuous first-order partial derivatives. Furthermore, all costs are assumed to be based upon a common unit of time. For example, if costs are on an annual basis, and if the end products are rented, $\Upsilon(A)$ is the annual rental costs of having A operating and nonoperating end products. If the end products are acquired for a fixed cost per end product, then $\Upsilon(A)$ is simply the equivalent uniform annual acquisition costs incurred during the life of an end product, multiplied by the number of end products acquired. In both cases, $\Upsilon(A)$ is a monotonically increasing function of A.

We assume that the end products are retired (disposed of) at the end of a fixed length of time and replaced by new end products. This length of time is therefore not directly related to the time which the end products spend in operation. Otherwise, we would have to consider that over a given time period, end products with shorter service times need to be replaced more often than end products with longer service times, and, in this case, $\Upsilon(A)$ becomes a function also of K. The assumption of a fixed end-product life arises from the empirical problem discussed later, where military aircraft are retired mainly because of obsolescence rather than deterioration.

The end-product tour length, H', is the number of days from the completion of an end-product repair to the time the end product returns for service. We assume that the tour length is fixed. This condition arises as a result of safety considerations, technological restrictions, contractual arrangement, or legal regulations.

If the tour length is not fixed, determination of the optimal values of the decision variables is complicated considerably. This occurs primarily because the number of reworks to be accomplished per unit time is independent of the relative values of A and K, for a given operating level, only if the tour length cannot be changed.† If the tour length is a decision variable, then the arrival rate is a function of the variable tour length. In this case, determination of the optimal values of A, K, and H involves consideration of the effect of different arrival rates on cost as well as consideration of the cost associated with repairing end products of different quality.

When the operating level is given (denoted by N''), the end product service-time cost function is represented by $\Phi(K, N'')$. It is assumed that $d[\Phi(K, N'')]/dK$ is less than zero. Otherwise we could reduce the service time and therefore increase the operating level without increasing rework costs.

The analysis is conducted in terms of a comparative static framework. There is assumed to be a

---

*Of course, the firm may incur other costs as well, but they will not be relevant in determining the optimal values of A and K. In particular are those costs generally related only to the operating level, rather than to the total number of end products. Examples are fuel consumption, emergency repair, and accident costs incurred during the operating tour.

†Cases in which the tour length is not fixed are developed in [7].

continuing need for the operating end products over an indefinite time period. A finite time-horizon model can be easily constructed by redefining the costs to be based upon the finite period. However, this approach will not ensure an optimal solution beyond the stated period. Cases in which multiperiod, multilevel end product requirements are specified are not considered here.

We start by noting that the total number of days that end products will be operational in a repair cycle of $K+H'$ days may be equivalently expressed as

$$N(K+H')=AH'.$$

Rearranging terms, we have

$$N/H'=A/(K+H'), \tag{1}$$

which represents the daily arrival rate of end products for repair. We can rewrite Equation (1) in the form

$$N=A-AK/(K+H'), \tag{2}$$

where N is constant as long as the arrivals are distributed uniformly over the repair cycle.

The second term on the right-hand side of Equation (2) represents the number of end products in repair. If the arrival rate is not constant, then this term can be interpreted as the average number of end products in repair. This average value is not influenced by the distribution of arrivals in the time period $K+H'$, for if we know that during a repair cycle all end products must arrive for service, then the average number of end products being repaired is independent of the distribution of arrivals.

If the service time is not deterministic, we can still calculate the average operating level by subtracting from A the average number of end products in repair. This latter quantity is equal to the average number of arrivals in a service time. Of course, if the service time is independent of the distribution of arrivals (for a given operating level), then the average number of non-operating end products is simply equal to the product of the average daily number of arrivals and the expected service time.

Cases in which the service time is not independent of the distribution of arrivals are beyond the scope of this paper. Future work along these lines may draw upon several papers that derive measures for the states of the system for a related class of "closed loop" queuing models. A two-stage model (operation and repair) has been studied by Koenigsberg [5]. Gordon and Newell [4] have studied a multistage model. Posner and Berholtz have written several papers on the subject, the most recent [6] dealing with the case of different classes of units being in the system.

The objective of this section is to develop necessary conditions for a minimum of the cost function

$$\Upsilon(A)+\Phi(K, N''),$$

subject to the condition that

$$N''=A-AK/(K+H').$$

To solve this constrained minimization problem, we form the function

$$C_\xi = \Upsilon(A) + \Phi(K, N'') + \xi[N'' - A + AK/(K + H')],$$

where ξ is the unknown Lagrange multiplier. Setting the partial derivatives of C_ξ with respect to A, K, and ξ equal to zero and simplifying, we find

$$A = N''(K + H')/H', \tag{3}$$

and

$$K = -(\Upsilon_A A + \Phi_K H')/\Phi_K, \tag{4}$$

where

$$\Upsilon_A = \frac{d[\Upsilon(A)]}{dA}$$

and

$$\Phi_K = \frac{d[\Phi(K, N'')]}{dK}.$$

Noting the arrival rate expressions in Equation (1), the appropriate root of K in Equation (4) is found by solving the following equation for K:

$$\Phi_K = -\frac{\Upsilon_A N''}{H'}. \tag{5}$$

Throughout the paper we will assume that second-order conditions are satisfied (see Debreu [2]).

The above method of solving the constrained minimization problem involves the use of an unknown Lagrange multiplier. The Lagrange multiplier can, in fact, be interpreted as the marginal cost of N. Thus, if the marginal revenue of N is known, then profits are maximized when N is chosen such that ξ is equal to the marginal revenue at N (see Bellman and Dreyfus [1]).

This section has shown how to calculate the level of the decision variables in order to minimize the cost of obtaining a given level of operating end products. If the firm is currently in operation, so that the decision variables are therefore already at some positive levels, we would then calculate the changes in these present levels which are necessary to move from the current to the optimal levels.

In some cases, however, it may be desirable to include the current levels of the decision variables directly into the optimizing problem. This situation is considered in the next section.

DEVELOPMENT OF THE INCREMENTAL RELATIONSHIPS

Assume that the current or initial levels of the decision variables are greater than zero. The following symbols are used:

A_0—the initial number of end products;
ΔA—the increase in the number of end products;
K_0—the initial number of days needed to service the end products;
ΔK—the reduction in the number of days needed to service the end products;

N_0 — the initial number of operating end products;

ΔN — the increase in the number of operating end products;

$v_0(\Delta A)$ — the cost of having ΔA additional end products when the initial number of end products is A_0, i.e., $v_0(\Delta A) = \Upsilon(A) - \Upsilon(A_0)$; and

$\phi_0(\Delta K, \Delta N)$ — the cost of reducing the end-product service time by ΔK days, when the initial service time is K_0 days, and of increasing the operating level of end products by ΔN, when the initial level is N_0, i.e., $\phi_0(\Delta K, \Delta N) = \Phi(K, N) - \Phi(K_0, N_0)$.

This section presents an alternative procedure for determining the minimum cost combination of the decision variables, A and K, in order to achieve a given operating level of end products. In the present method, the decision variables are defined* as positive changes in A and negative changes in K. We then seek the values of these changes which minimize the periodic cost of obtaining a given increase in the operating level of end products.

One reason for considering this second approach is to note the symmetry of the solutions obtained from both methods. A more practical reason is that in the application of the model, it is sometimes preferable to work with incremental forms of the functions. This will be demonstrated later.

The problem addressed here is this: find the increased number of operating and non-operating end products, ΔA, and the reduction in the end-product service time, ΔK, in order to obtain a given increase in the operating level of end products at the least periodic cost. We observe from Equation (2) that if the service time is reduced by ΔK days and the number of operating and non-operating end products is increased by ΔA, the increase in the operating level, ΔN, is given by

$$\Delta N = [\Delta A H'/(H' + K_0 - \Delta K)] + A_0 \Delta K H'/[(H' + K_0)(H' + K_0 - \Delta K)]. \tag{6}$$

Hence, the problem just discussed can be expressed as follows:

$$\text{Minimize } v_0(\Delta A) + \phi_0(\Delta K, \Delta N''), \tag{7}$$

subject to the condition that

$$\Delta N'' = [\Delta A H'/(H' + K_0 - \Delta K)] + A_0 \Delta K H'/[(H' + K_0)(H' + K_0 - \Delta K)]. \tag{8}$$

To minimize expression (7), subject to expression (8), we form the function

$$C_{\xi,0} = v_0(\Delta A) + \phi_0(\Delta K, \Delta N'') + \xi[[\Delta N'' - \Delta A H'/(H' + K_0 - \Delta K)] - A_0 \Delta K H'/ [(H' + K_0)(H' + K_0 - \Delta K)]]. \tag{9}$$

Setting its partial derivatives equal to zero and simplifying yields

*It is assumed that it is not beneficial or feasible to reduce the existing level of resources, i.e., $\Delta K \geq 0$, $\Delta A \geq 0$. In cases where this restriction is violated and it is beneficial and feasible to reduce the level of a resource, then a multi-stage approach is required if the savings functions from the reduction differ from the resource cost functions.

$$\Delta A = [\Delta N''(H' + K_0 - \Delta K)/H'] - A_0 \Delta K/(H' + K_0), \tag{10}$$

and

$$\Delta K = [-v_a(A_0 + \Delta A)/\phi_k] + H' + K_0, \tag{11}$$

where

$$v_a = \frac{d[v_0(\Delta A)]}{d(\Delta A)},$$

$$\phi_k = \frac{d[\phi_0(\Delta K, \Delta N'')]}{d(\Delta K)},$$

and

$$\phi_0(\Delta K, \Delta N'') = \Phi(K, N'') - \Phi(K_0, N_0).$$

Finally, solving Equation (11) for the appropriate root of ΔK, we observe the relationship to be symmetrical to Equation (5):

$$\phi_k = \frac{v_a(N_0 + \Delta N'')}{H'}. \tag{12}$$

Thus, the condition for optimal resource allocation in both approaches to the problem can be equivalently stated as follows: equate the absolute value of the marginal factor cost of repair services divided by the marginal factor cost of end products, to the arrival rate of end products for repair. This condition applies to many variations of the basic model presented in this paper, and one of these variations is studied in the next section.

APPLICATION OF THE MODEL TO THE NAVY AIRCRAFT PIPELINE PROBLEM

The model is now applied to the case of the repair of Navy combat aircraft. In the economic analysis of a military weapon system, such as Navy combat aircraft, it is convenient to divide the system into an operating status and a logistic status. Aircraft in an operating status are either in use or at least available for operations. Aircraft in a logistic status are not available to operating commanders because they are either being repaired at a rework facility or they are in transit. These latter aircraft are known as "pipeline" aircraft.

Recently, there were 300 Navy F–4 aircraft being repaired or scheduled for repair at the North Island facility. These aircraft have a tour length of 450 days (21 months) and require 44 days to service. Based on this information, we calculate the operating level of F–4 aircraft from Equation (2) to be 273 planes.*

The decision to rework aircraft in a given number of days is generally based on the anticipated

*We are only concerned with scheduled repair in this paper. Therefore, aircraft that may be down for unscheduled repair are still referred to as "operating."

workload; that is, the number of arrivals per unit time for the given tour length, and not based on the total number of aircraft, A. After the aircraft service time is determined, the value of A is then chosen, taking the value of K as given. Hence, as a result of not determining the values of the decision variables simultaneously, A and K may not be optimal.

Therefore, we want to determine whether or not the mix of resources between aircraft and repair services was optimal for an operating level of 273 aircraft. To demonstrate that it was not, we will show that a first-order condition for optimal resource allocation is not satisfied.

At the current operating level, $\Delta N''$ is equal to zero. So from Equation (12), this first-order condition can be written:

$$\phi_k = \frac{v_a A_0}{H' + K_0}. \tag{13}$$

We will show that this condition did not, in fact, prevail.

The following assumptions are made:

1. The acquisition cost of an F–4 aircraft is \$3 million.
2. The life of an F–4 aircraft and the reparable spare parts which comprise the aircraft is 10 years.
3. The arrival rate of F–4's at North Island is three aircraft per week.
4. The work content and the workload of scheduled F–4 repairs at North Island will not change, materially, during the life of the weapon system.
5. Aircraft currently in use will be replaced, upon retirement, by new aircraft.
6. All daily time units are in terms of work days of five per week.
7. Additional resources which can be applied to reduce the aircraft service time are limited to recoverable items.
8. For the feasible workloads to be accomplished, the aircraft service time is independent of the distribution of arrivals.

It is assumed that only the assembly stage of the aircraft repair can be reduced by the addition of factors of production, and these factors are limited to reparable spares. During this stage of the aircraft service, reparable items which have been removed from the aircraft are sent to rework shops for repair and then reinstalled on the aircraft. The assembly stage usually represents about 50 percent of the aircraft service time.

We have found that during the period under study, the assembly stage of 27 F–4's serviced at North Island averaged 28 days, which is 16 days in excess of the planned time. It has been determined that 11 of this 16-day delay can be attributed to the absence of spare stocks. Using a technique developed in [9], the (average) reduction in the aircraft service time (delay) associated with increasing the stock of recoverable spares has been calculated. Table 1 indicates the investment in spares necessary to achieve specified reductions in the aircraft service time.

Now, if the existing 44-day aircraft service time was optimal, then Equation (13) will be satisfied for $\Delta K = 0$. When ΔK is integer valued, this first-order condition becomes

$$\phi_0(\Delta K, \Delta N'' = 0) - \phi_0(\Delta K - 1, \Delta N'' = 0) \leq \frac{v_a A_0}{H' + K_0} \leq \phi_0(\Delta K + 1, \Delta N'' = 0) - \phi_0(\Delta K, \Delta N'' = 0). \tag{14}$$

TABLE 1. *Aircraft days saved by adding spares*

Items to acquire		Repairable spare cost		Aircraft days in delay		
Total	Marginal	Total (thousands)	Marginal (thousands)	Existing delay	Reduction in delay	Total saving
0		0		11		
9	9	$92	$92	10	1	1
31	22	2,101	2,009	5	5	6
46	15	2,975	874	3	2	8
72	26	3,426	451	2	1	9
129	57	3,941	515	1	1	10
504	375	5,585	1,644	0	1	11

From columns three and seven of Table 1, the end-product service time cost function can be represented as follows:*

Service days reduction (ΔK)	Cost in thousands $\phi_0(\Delta K, \Delta N'' = 0)$
1	$92
6	2,101
8	2,975
9	3,426
10	3,941
11	5,585

Furthermore, we stated previously that the following information was ascertained:

$v_a = \$3$ million (per aircraft),
$A_0 = 300$ (aircraft),
$K_0 = 44$ (days), and
$H' = 450$ (days).

When these figures are substituted into condition (14), we find that, because of the limiting values which ΔK can assume, there is no value of ΔK which satisfies (14). Indeed, in this case, the optimal value of ΔK occurs at the end point of admissible values, which is 11 days.

Hence, this implies that a service time of 44 days was not optimal and should, in fact, have been 33 days. To support an operating level of 273 aircraft, this means that we only needed 293 operating and non-operating aircraft. Therefore, if we could have disposed of seven F–4 aircraft at their acquisition price, then these aircraft are "worth" $21 million. Since the additional repairable spare cost required to achieve a service time of 33 days is, from Table 1, $5.585 million, the annual "saving" which would have accrued to the Navy is $1.542 million.

Although we cannot, in fact, dispose of these aircraft, they could have been sent to a different repair facility for maintenance, and thus reduced further planned F–4 acquisitions by seven aircraft.

*Costs and benefits are in terms of a 10-year period, unless stated otherwise. The aircraft and repairable spares costs represent their acquisition prices.

In this instance, the annual saving of \$1.542 million was still possible by substituting transferred aircraft released from repair for new acquisitions.

Suppose that this substitution was not a feasible alternative. A reasonable strategy to pursue, if the total number of aircraft is fixed, is to invest in repair resources up to the point at which the marginal cost of increasing the operating level by reducing the servicing time, C_K, is equal to the marginal cost of increasing the operating level by acquiring aircraft, C_A, where*

$$C_A = \frac{\dfrac{\partial C_\eta}{\partial A}}{\dfrac{\partial N}{\partial A}}; \qquad C_K = \frac{\dfrac{\partial C_\eta}{\partial K}}{\dfrac{\partial N}{\partial K}};$$

and

$$C_\eta = \Upsilon(A) + \Phi(K, N). \tag{15}$$

Suppose that the costs of increasing the operating level, for a given service time, are proportional to the increase in the operating level. In this case we can replace

$$\Phi(K, N) = \Phi(K, N_0)\frac{N}{N_0} \tag{16}$$

in equation (15). Substituting the appropriate terms for C_A and C_K, equating these expressions, and simplifying, we have

$$-\frac{\partial[\Phi(K, N_0)]}{\partial K} = \frac{\Upsilon_A A_0}{H' + K_0}.$$

Interestingly enough, this is the same first-order condition obtained in Equation (13) and condition (14). Of course, the appropriate value of $K(\Delta K)$ already has been found to be 33 (11) days.

The 11-day reduction in service time, costing \$5.585 million (which excludes the cost of the additional reworks that must be accomplished as the operating level increases) would have increased the operating level from 273 to 279 planes. This is in lieu of increasing the operating level by six planes as a result of acquiring seven additional F–4's, at a cost of \$21 million. (This figure also excludes the cost of the added reworks.)

The cost of the additional reworks, when the service time is reduced, arises from both the added number of arrivals per unit time, and from the reduced service time of these additional arrivals. These costs can be denoted by the difference between the total repair cost of reducing the service time, given by

$$\Phi(K, N) - \Phi(K_0, N_0),$$

*The dynamics of the adjustment process of the operating level to changes in the service time are discussed in [8] and in the appendix.

and the cost of decreasing the service time for the initial operating level only, given by

$$\Phi(K, N_0) - \Phi(K_0, N_0).$$

From this definition and from Equation (16), these costs can be rewritten as

$$\Phi(K, N) - \Phi(K, N_0) = \Phi(K, N_0) \frac{N - N_0}{N_0}. \tag{17}$$

On the other hand, the cost of the additional reworks, when the number of aircraft is increased, results from simply having more arrivals per unit time. It is easy to demonstrate that this cost can be expressed by

$$\Phi(K_0, N) - \Phi(K_0, N_0) = \Phi(K_0, N_0) \frac{N - N_0}{N_0}. \tag{18}$$

To calculate the costs represented by equations (17) and (18), we note that, during the period under study, it was determined by Fiekowsky and Schwartz [3] that the (variable) cost of repairing an F–4 aircraft at North Island in 44 days was \$100,000 per repair. Each plane will undergo five reworks during its 10-year life, so that with 300 total aircraft, there will be 1,500 F–4 reworks performed at North Island in a 10-year period. Accordingly, the then existing repair cost, $\Phi(K_0, N_0)$, is estimated at \$150 million.

Furthermore, we have already calculated the cost of reducing the service time by 11 days to be \$5.585 million, if the operating level is not changed. Therefore, $\Phi(K, N_0)$ for $K = 33$ is equal to \$155.585 million.

Substituting the value of $\Phi(K_0, N_0)$ into Equation (18) yields the additional rework cost incurred as the result of acquiring seven more aircraft, which is \$3.297 million. Since the acquisition cost of these aircraft is \$21 million, we find the total cost of increasing the operating level from 273 to 279, by acquiring seven aircraft, was \$24.297 million.

Replacing $\Phi(K, N_0)$ with \$155.585 million in Equation (17), we find the additional rework costs incurred if the service time was reduced by 11 days to be \$3.420 million. Adding to this figure the cost of reducing the service time of the original operating level, \$5.585 million, we find that the total cost of reducing the service time from 44 to 33 days was \$9.005 million.

Therefore, reducing the service time to increase the operating level by six aircraft costs \$15.292 million less than acquiring seven aircraft to obtain the same increase in the operating level. On an annual basis, it can be seen that this "saving" is almost equivalent to the savings obtainable in the case of aircraft transferability. These findings have added significance because the Navy has traditionally increased the operating level by acquiring more aircraft.

SUMMARY

This paper has shown how to determine the minimal cost combination of end products and repair resources to maintain a given number of operating end products. Clearly, if the firm can determine the minimum cost of achieving different levels of operating end products, and if the firm can also determine the revenue associated with each operating level, then it can find the operating level which will maximize its profits.

The model has been applied to the case of F–4 aircraft reworked by the Navy. It was shown that the mix of resources was not appropriate to support the then existing operating level. It was determined that an annual saving of over one and one-half million dollars could have been achieved if corrective action, as specified by the model, had been implemented.

Appendix

DYNAMICS OF THE END-PRODUCT OPERATING LEVEL ADJUSTMENT PROCESS

The arguments in the preceding sections of this paper were in terms of a comparative static framework. That is, we have calculated the effect of changes in the service time on the operating level of end products without considering the way in which the adjustment occurs. Knowledge of the adjustment process is significant, because when the operating level changes, the workload to be accomplished and the distribution of end-product arrivals are also affected. Therefore, if the firm can determine the adjustment process, it can then allocate its resources more efficiently by anticipating fluctuations and changes in the workload. This section is concerned with describing the adjustment process, through time, of the operating level and hence the number of end products in repair. We assume that an increase occurs in the operating level as a result of a reduction in the end-product service time, but that the total number of end products is fixed.

One of the difficulties encountered in describing the dynamics of the situation is that the adjustment in the number of end products in repair through time is related not only to the number of days by which the service time is reduced, but also to (1) the number of end products in repair at the time the service time is reduced; (2) the distribution of end-product arrivals (which may not be independent of the number of end products currently in repair); and (3) the manner in which the reduction in service time affects those end products which are being processed when the reduction takes place. To simplify the analysis we make the assumption that prior to reducing the end-product service time, the arrivals of the A_0 end products are distributed uniformly over a repair cycle of $H' + K_0$ days. Therefore, the number of end products in repair is equal to the number of arrivals in K_0 days, which is equivalent to $A_0K_0/(H' + K_0)$.

We also assume that the service time of end products which arrive for repair by the close of an arbitrary day t is $K_0 - \Delta K$ days, and those end products which are in repair at the close of day t require K_0 days to service. Then from (the close of) day t to (the close of) day $t + K_0 - \Delta K$, the number of departures is equal to $A_0(K_0 - \Delta K)/(H' + K_0)$, where $A_0/(H' + K_0)$ is the arrival rate of end products. In this interval, departing end products consist solely of end products which require K_0 days to service, and the number of end products in repair remains at Λ_0K_0, where Λ_0 is equal to $A_0/(H' + K_0)$.

During the time $t + K_0 - \Delta K$ to $t + K_0$, the number of departures is double the number of arrivals, since end products requiring both K_0 days and $K_0 - \Delta K$ days to repair are completed. During these ΔK days, the number of end products in repair declines at a rate equal to Λ_0, and at the close of day $t + K_0$ there are $\Lambda_0(K_0 - \Delta K)$ end products in repair. Thus, it takes K_0 days for the number of end products in repair to be reduced by $\Lambda_0\Delta K$ end products.

The number of end products in repair remains at $\Lambda_0(K_0 - \Delta K)$ until time $t + H' + K_0 - \Delta K$, or for $H' - \Delta K$ days, when end products processed in $K_0 - \Delta K$ days begin to return for servicing. From time $t + H' + K_0 - \Delta K$ to $t + H' + K_0$, arrivals of end products are double the departures, and the number of end products in repair grows at a rate equal to Λ_0. By time $t + H' + K_0$, the number of end products in repair is once again equal to Λ_0K_0, and it remains at this level for $K_0 - 2\Delta K$ days (if K_0 exceeds

$2\Delta K$), since during this period the arrival rate, Λ_0, is equal to the departure rate. Then, starting at the end of time $t+H'+K_0$, the number of departures is double the number of arrivals for ΔK days, and there will again be $\Lambda_0 K_0$ end products in repair by time $t+H'+2K_0-\Delta K$. A diagramatic representation of the adjustment process for the case where K_0 exceeds $2\Delta K$ is shown in Figure 1.

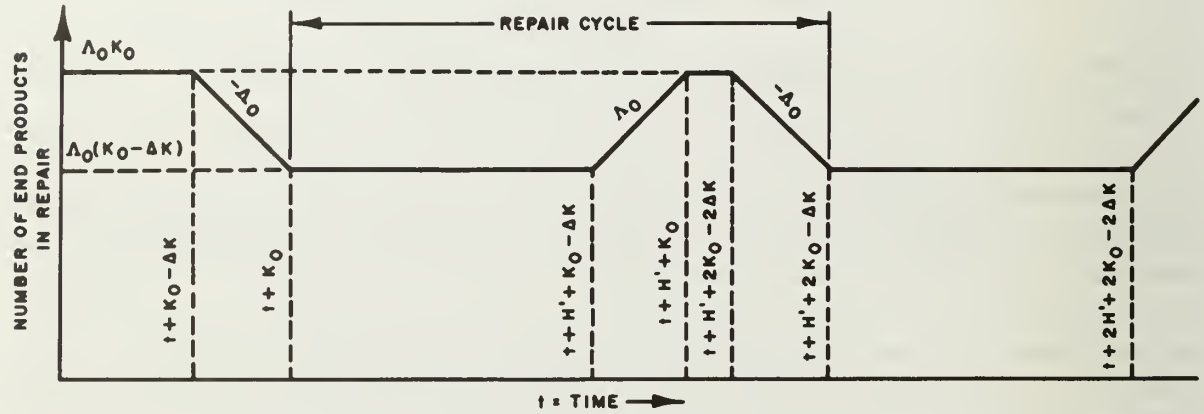


FIGURE 1. Dynamic adjustment process for end products in repair when the initial service time exceeds twice the change in the service time.

In the case where $2\Delta K$ is equal to or greater than K_0, it still takes K_0 days for the $\Lambda_0 K_0$ end products in repair to be reduced to $\Lambda_0(K_0-\Delta K)$, and the number of non-operating end products will remain at this level until the close of day $t+H'+K_0-\Delta K$. At this time, the end-product arrival rate is double the departure rate, and the number of end products in repair increases at a rate equal to Λ for $K_0=\Delta K$ days. Then, at time $t+H'+2K_0-2\Delta K$, both the arrival rate and the departure rate are equalized at $2\Lambda_0$, and there will be $2\Lambda_0(K_0-\Delta K)$ end products in repair for $2\Delta K-K_0$ days. At time $t+H'+K_0$, the number of departures is double the number of arrivals, and the number of end products in repair declines at a rate equal to Λ_0 for $K_0-\Delta K$ days. By time $t+H'+2K_0-\Delta K$, there are once again $\Lambda_0 K_0$ end products in repair, and the number of non-operating end products remains at this level until time $t+2H'+2K_0-2\Delta K$. A diagramatic representation of the adjustment process for the case where $2\Delta K$ is equal to or greater than K_0 is shown in Figure 2.

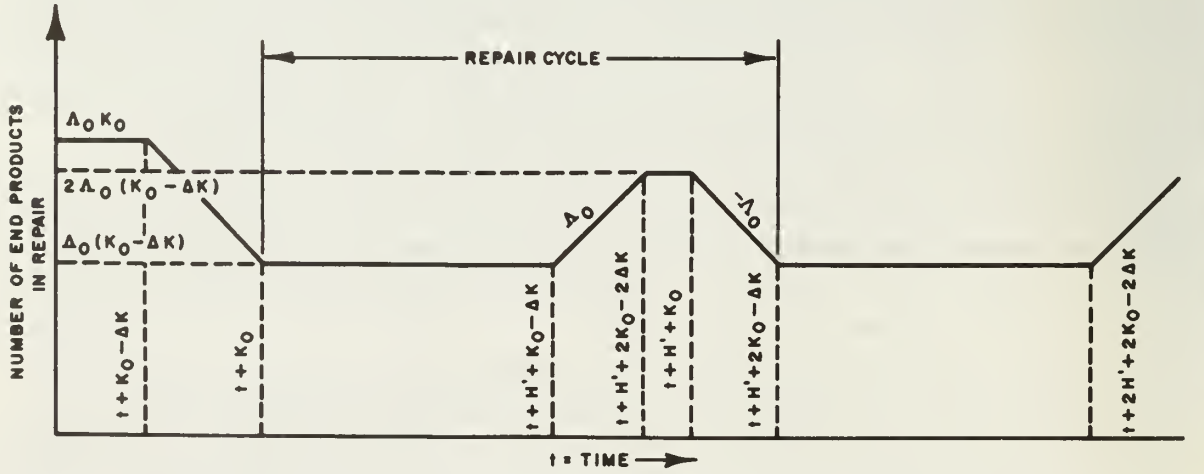


FIGURE 2. Dynamic adjustment process for end products in repair when the initial service time is equal to or less than twice the change in the service time.

REFERENCES

[1] Bellman, R. and S. Dreyfus, *Applied Dynamic Programming* (Princeton University Press, Princeton, New Jersey, 1962).

[2] Debreu, G., "Definite and Semi-Definite Quadratic Forms," Econometrica, *XX*, *295–300* (Apr. 1952).

[3] Fiekowsky, S. and A. Schwartz, "Cost of Aircraft Depot Maintenance: The F–4 Case," Proceedings of the Second Annual Department of Defense Cost Research Symposium (Mar. 1967), p. C–11.

[4] Gordon, W. and G. Newell, "Closed Queuing Systems with Exponential Servers," Operations Research *15*, 254–265 (1967).

[5] Koenigsberg, E., "Finite Queues and Cyclic Queues," Operations Research, *8*, 246–253 (1960).

[6] Posner, M. and B. Bernholtz, "Closed Finite Queuing Networks with Time Lags and with Several Classes of Units," Operations Research, *16*, 977–985 (1968).

[7] Rose, M. and L. Brown, "An Incremental Production Function for the End-Item Repair Process," AIIE Transactions (June 1970), pp. 166–171.

[8] Rose, M., "An Investment Model for Reparable Assets: The F–4 Case," Research Contribution 31 (August, 1969), Institute of Naval Studies, Center for Naval Analyses, Arlington, Virginia.

[9] Rose, M., "Computing the Expected End-Product Service Time Using Stochastic Item Delays," Operations Research *19*, 524–540 (Mar.–Apr., 1971).

REQUIREMENTS OF AN "OPTIMIZER" FOR COMPUTER SIMULATIONS*

Dennis E. Smith

HRB-Singer, Inc.
State College, Pennsylvania

ABSTRACT

Computer simulation has many advantages. However, one major disadvantage is that, in all too many cases, the attempt to use computer simulation to find an optimum solution to a problem rapidly degenerates into a trial-and-error process. Techniques for overcoming this disadvantage, i.e., for making optimization and computer simulation more compatible, are applicable at two points in the development of the overall computer simulation. Techniques which are used within actual construction of the mathematical models comprising the simulation will be labeled as internal methods, while those which are used after the simulation has been completely developed will be termed external methods.

Because external methods appear to offer the largest potential payoff, discussion is restricted to these methods, which are essentially search techniques. In addition, the development of an "Optimizer" computer program based on these techniques is suggested.

Although drawbacks to the use of search techniques in the computer simulation framework exist, these techniques do offer potential for "optimization." The modification of these techniques to satisfy the requirements of an "Optimizer" is discussed.

I. INTRODUCTION

Computer simulation has become an extremely useful and popular method of operations research. The reasons for this usefulness and popularity are many. Computer simulation can be used in those situations where the problem under investigation is too complex to be handled analytically. Because random variables and their associated probability distributions may be directly incorporated into the simulation, more realism is permitted by computer simulation than by analytic methods. The development of a computer simulation and its associated mathematical models does not, except in special cases, require more than a modicum of mathematical knowledge, when compared with the application of analytical techniques. The results of a computer simulation study are, in general, more easily "sold to" management than are the results of an analytical study, because ". . . decision makers, being operationally oriented, tend to put more faith in the results of a simulation than in the results of mathematical abstractions because the former frequently appear to be directly simulating operational procedures" (Koopman and Berger [16]).

It should be noted that the terms "mathematical model" and "computer simulation" mean many things to many people. To provide a common basis, a mathematical model will be defined as a logically derived set of formulas, equations, and probability distributions which represent the relationships between the various variables in the situation being considered. Computer simulation will be defined

*The study on which this paper is based was supported by the Office of Naval Research under Contract No. N00014-69-C0284, Task No. NR364-010.

as the running (i.e., the application) of a mathematical model which is in the form of a computer program or a collection of computer programs* to obtain answers to questions of interest.

A common type of computer simulation is one based on a probabilistic model characterized by an assemblage of mathematically related random variables. Such a model in computer program form relies on the use of sequences of computer-generated pseudo-random numbers, in conjunction with the appropriate probability distributions, to select specific values of the random variables. It is with computer simulations involving this type of probabilistic model† that this report is primarily concerned.

As noted previously, computer simulation has many advantages. However, one major disadvantage is that, in all too many cases, the attempt to use computer simulation to find an optimum solution to a problem rapidly degenerates into a trial-and-error process. This disadvantage, which has, unfortunately, received relatively little attention in the literature, is addressed in the present study.

Techniques for overcoming this disadvantage, i.e., for making optimization and computer simulation more compatible, are applicable at two points in the development of the overall computer simulation. Techniques which are used within actual construction of the mathematical models comprising the simulation will be labeled as internal methods, while those which are used after the simulation has been completely developed will be termed external methods. Internal methods involve modification of the mathematical models in order to make them more efficient from the standpoint of potential for optimization.

After development of the overall computer simulation is completed (i.e., after the models have been constructed and programmed for the computer), the simulation itself may be regarded as an experimental situation in which statistical experimental designs are pertinent. External methods involve the use of search techniques which are based on experimental designs. The simulation situation, however, poses some very difficult problems in the use of such techniques and, in fact, has been said to ". . . render insufficient some of the classical principles of experimental design . . ." (Jacoby and Harrison [15]).

Although internal methods are actually built into the simulation when the mathematical models are constructed, external methods, on the other hand, do not affect the inner workings of the simulation, and are essentially independent of model construction. Because of this, external methods appear to have wider applicability than internal methods. Hence, this paper confines discussion to external methods only, and suggests the development of an "Optimizer" computer program based on these methods. This "Optimizer" would use the inputs and outputs of the simulation plus additional information from the simulation user to automate, to a degree, the optimization process and, hopefully, provide a great improvement over trial-and-error.

II. COMPUTER SIMULATION AS A "BLACK BOX"

A computer simulation may quite realistically be regarded as involving a "black box" in which the values of input variables are combined in some manner to produce output variables or responses. These responses are usually figures of merit or measures of effectiveness of some type. Figure 1 summarizes this "black box" view of computer simulation when only one output variable (response) is involved.

As will be noted from this figure, the input variables are of two types: (1) controllable factors,

*Mathematical models in this form are sometimes referred to as "simulation models."

† Such a model is often referred to as a Monte Carlo model.

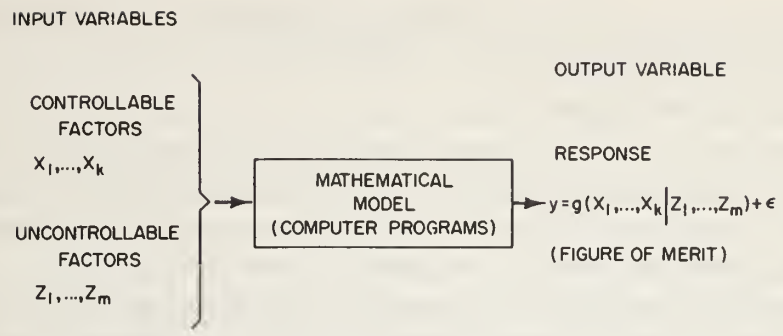


FIGURE 1. A black box view of computer simulation.

and (2) uncontrollable factors. The controllable factors are those input variables having values which can be controlled in the "real world." For example, input variables such as submarine speed, aircraft altitude, and destroyer bearing pertaining to United States Naval forces would be controllable factors because their values could be varied at will by the appropriate decision maker in the Naval chain of command. On the other hand, these input variables (submarine speed, aircraft altitude, and destroyer bearing) relating to enemy forces must be regarded as uncontrollable factors because the U.S. Navy cannot vary the values of these factors at will. In addition, environmental characteristics such as sea state, temperature, and amount of precipitation must also be regarded as uncontrollable factors because their values, too, cannot be controlled.

As Figure 1 indicates, the computer simulation "black box" produces the response $y=g(X_1, \ldots, X_k|Z_1, \ldots, Z_m)+\epsilon$ which is a function of both the controllable and uncontrollable factors. Because of the statistical variation incorporated into most computer simulations, however, the observed response y varies even if two identical situations are run, i.e., if all input variables have the same values both times, the simulation is used to produce a response.* This difference is represented by the term ϵ, which, in statistical terms, may be regarded as sampling error.

A. "Optimization" in Simulation

In many cases, an important goal of computer simulation is to determine the optimum solution to the problem under consideration. In other words, it is desired to know which values of the controllable factors produce the optimum response or figure of merit. Unfortunately, the lack (or complexity) of an explicit mathematical expression of the problem in a simulation framework generally precludes the application of the standard methods of optimization (e.g., the calculus of variations, Lagrange multipliers, partial differentization; mathematical programming techniques) which prove valuable in an analytic framework. Thus, in the simulation framework one often sees a trial-and-error approach to optimization.

Before pursuing the topic further, it should be noted that optimization, per se, in simulation is an ultimate goal which in practical situations is usually unreachable. Ordinarily, the most that one can hope to accomplish given the constraints of limited resources (i.e., time and funds) is the location of an improved, or a "good" solution. That is, the result may be the identification of a solution which, while not necessarily optimal, is acceptable and better than any previously available solution. Thus, for

*If different sequences of pseudo-random numbers are used in each of the "identical" computer runs.

purposes of the current study, the term "optimization" when placed in quotation marks will refer to a search for an improved, though not necessarily optimal, solution.

B. Assumptions About the Simulation

The focus of the current study is on "optimization" within that class of simulations where a single response (i.e., figure of merit or measure of effectiveness) exists, the controllable factors are quantitative and continuous (or at least approximately so), and the uncontrollable factors have known values or at least generate random variables with known probability distributions.

Thus, for example, controllable factors pertaining to speed, bearing, altitude, depth, time, and sensor parameters are the kind of factors assumed to be under investigation. On the other hand, discrete controllable factors pertaining to type of radar, type of torpedo, or type of sonar will not be considered in this paper unless a quantitative parameterization exists.

It should be noted that none of the controllable factors (the X_i's) and none of the uncontrollable factors (the Z_j's) are random variables. Rather, the values of these factors may or may not generate random variables. For example, depending upon the manner in which the simulation is constructed, an input value of $X_1 = 10$, where controllable factor X_1 pertains to submarine speed (in knots), may be considered as a fixed value (e.g., the actual submarine speed used in the simulation is fixed at $X_1 = 10$) or as generating a random variable (e.g., the actual submarine speed used in the simulation is a random variable arising from some specified probability distribution with mean, i.e., expected value, of $X_1 = 10$). Likewise, an input value of $Z_1 = 20$, where uncontrollable factor Z_1 determines the number of enemy submarines in a certain region of the ocean, can be considered as a fixed value (e.g., it is assumed in the simulation that there are exactly 20 enemy submarines in the region) or as generating a random variable having a known probability distribution (e.g., it is assumed in the simulation that there are k enemy submarines in the region, where the value of k is unknown, but where the probability distribution of k is a binomial distribution depending on two parameters, one of which is $Z_1 = 20$).

Regardless of whether the controllable and uncontrollable factors have fixed values or generate random variables, their use in the simulation provides, in general, a number of responses. Although most computer simulations give rise to many responses (figures of merit or measures of effectiveness), it will be assumed in this report that a single response (i.e., a composite figure of merit) is available as a weighted sum or some other function of the individual responses produced by the simulation.* Thus, only a single response need be considered. If such an overall figure of merit is not readily available for a particular problem, there is usually, in the absence of a weighting scheme for obtaining a composite figure of merit, one principal figure of merit which may serve as a single response.

In general, the assumptions discussed in this section, while excluding some simulation situations, are not exceedingly restrictive.

III. EXTERNAL METHODS

External methods for aiding in "optimization" do not affect development of the mathematical model or computer programs which compose the "black box" of Figure 1. External methods of "optimization" treat the computer simulation as an experiment in which the responses produced by different simulation runs may be observed, and prescribe decision rules to be followed in selecting

*The choice of a single response (composite figure of merit or overall measure of effectiveness) often is an extremely difficult problem. However, this problem is not within the purview of this paper.

the values of the controllable factors to be used in the simulation runs.

Under the assumptions in the preceding section, if there are k controllable factors in the simulation, the response (disregarding statistical variation) lies on a surface in $(k+1)$—dimensional space. This surface is referred to as the response surface. Figure 2 depicts a response surface where two controllable factors are involved. In this two-factor situation, the response surface may, in a certain sense, be regarded as a hill which is to be climbed. Although the topography of the response surface (hill) cannot be directly shown, it can be indicated by contours corresponding to various responses (elevations). The objective of optimization is, of course, to determine the location, i.e., the values of X_1 and X_2, of that point which produces the best response. This point is indicated in the figure.

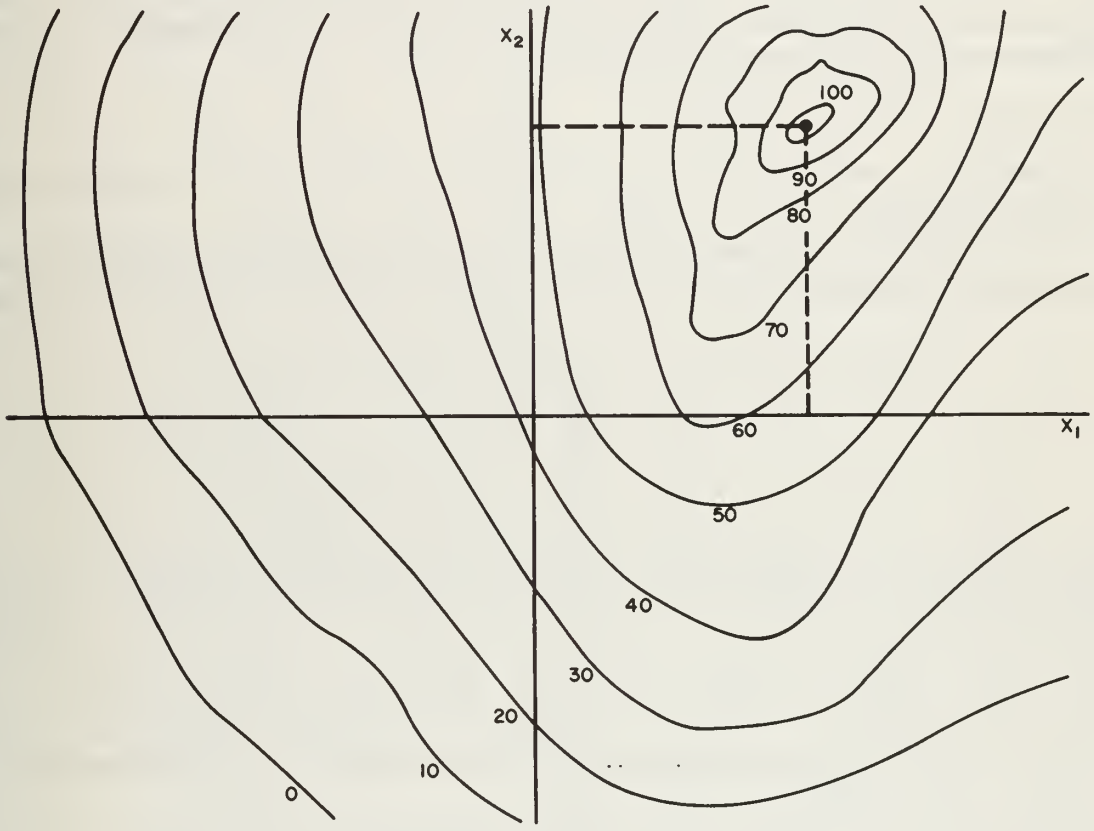


FIGURE 2. A response surface involving two factors.

External methods, then, are search techniques for exploring the response surface. Although a substantial amount of work [11, 14, 20] has been done on the development of search techniques, much of it is inappropriate in the simulation context because at least one of the following four assumptions is made:

(1) The optimization search involves a function which has known algebraic form.

(2) All responses are observed without error, i.e., without statistical variation.

(3) The number of available computer runs is essentially unlimited.

(4) The number of factors under investigation is relatively small, e.g., less than 10.

In general, none of the assumptions are valid for the computer simulation situation. The inappropriate-

ness of the first assumption has been discussed in section II. A., and the inappropriateness of the other assumptions will be mentioned in the following sections.

The suggested search techniques which have appeared in the literature and are pertinent in the simulation context may be classified as adaptive or nonadaptive. In the nonadaptive techniques, the observed response from a computer run does not influence the values of the controllable factors chosen for any succeeding run. Adaptive techniques, however, permit selection of the values of controllable factors for future computer runs to be based on the observed responses from previous runs.

Although different variations of search techniques have appeared in the literature [3, 5, 15, 21, 22], there are four primary techniques which serve as a basis for most others. These are: (a) factorial design, (b) random search, (c) single-factor approach, and (d) Response Surface Methodology (RSM). Of these, the factorial design and the random search are nonadaptive techniques, while the single-factor approach and Response Surface Methodology are adaptive techniques.

A. Nonadaptive Search Techniques

Two techniques, one based on a factorial design and the other based on a random search, are considered in this section.

1. Factorial Design—The factorial design, or factorial experiment, involves the selection of several values of each controllable factor. Simulation runs are made for all combinations of these factor values.

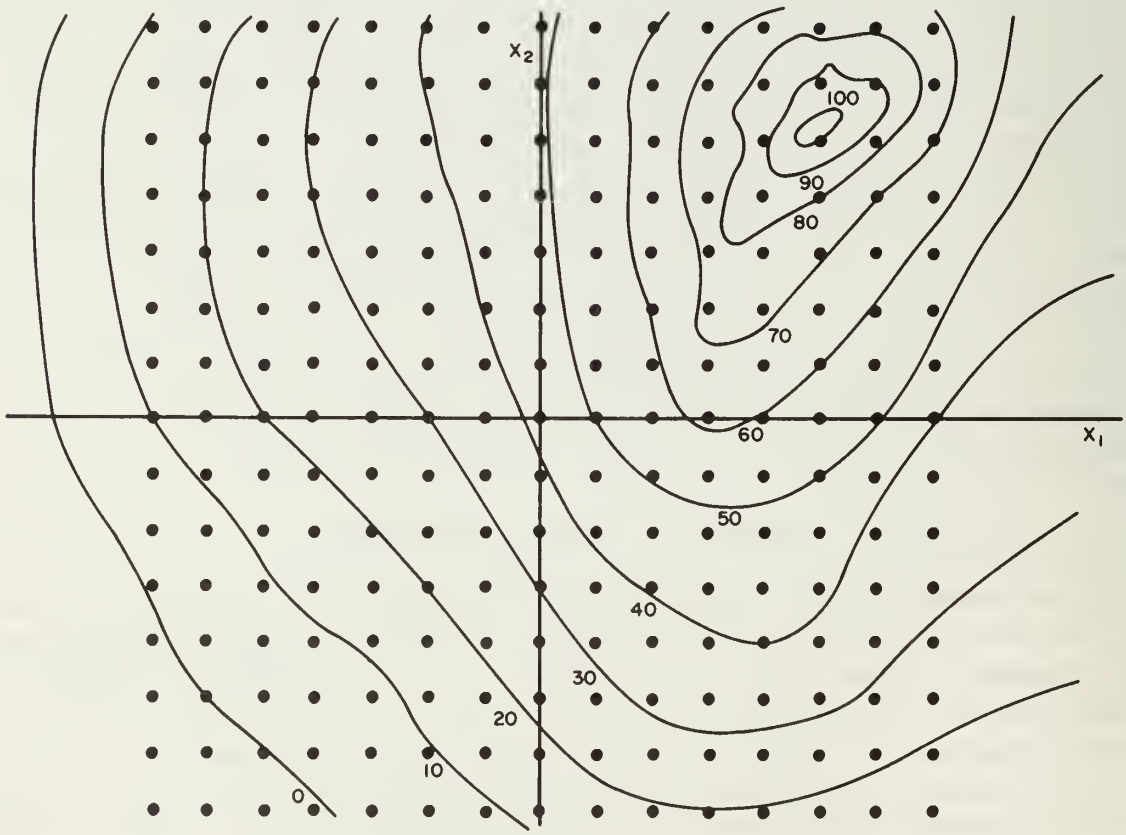


FIGURE 3. Application of a factorial design.

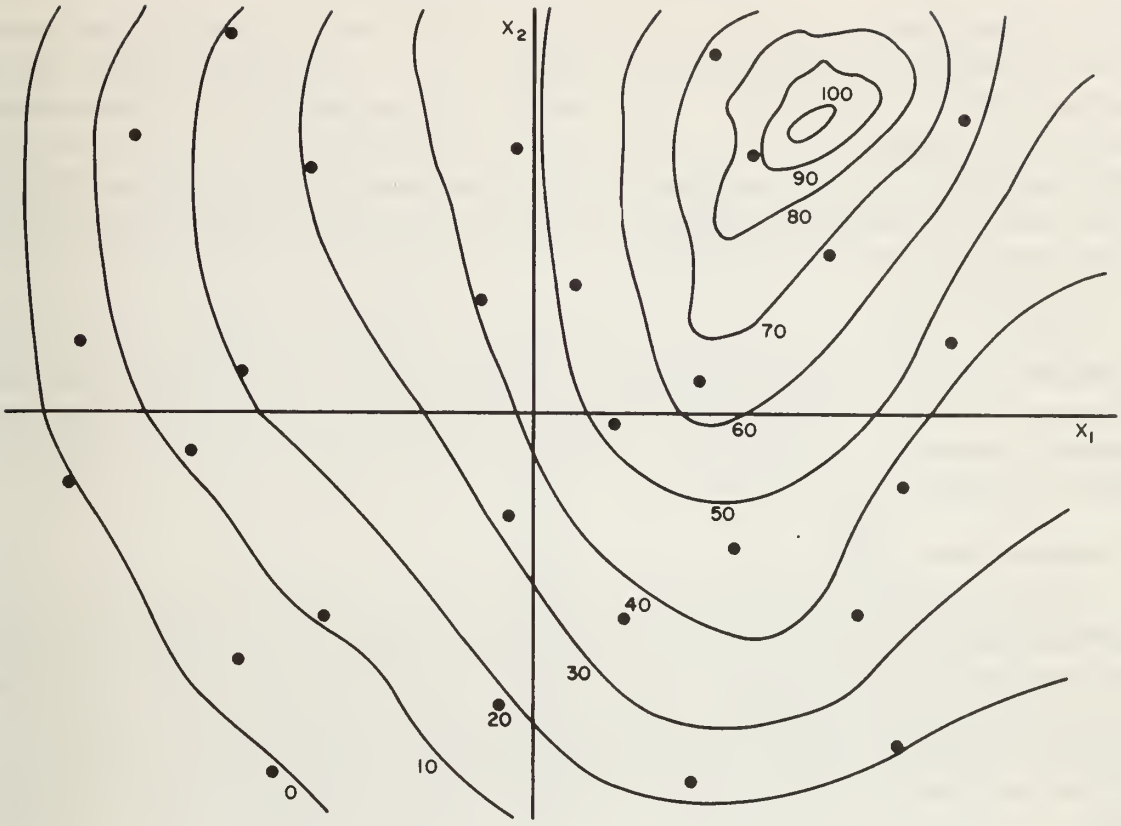


FIGURE 4. Application of a random search.

For example, if five values of each of three controllable factors were considered, a total of $5 \times 5 \times 5 = 125$ combinations would exist, and hence 125 simulation runs would be required. Figure 3 indicates a factorial design used to explore a response surface when there are only two controllable factors. As can be seen from the figure, the use of a factorial design can be regarded as laying down a grid of points corresponding in simulation runs which are to be made.

The factorial design can be used in two different ways to select "optimum" values of the controllable factors. One can simply select that point in the grid which gives the best results, or one can use a set of points in a region to fit a regression equation which can then be used to predict the location of an "optimum" point. Regardless of the way in which the "optimum" is to be selected, it should be noted that a major advantage of the factorial design is that it provides a thorough, systematic search of the response surface. However, this thoroughness itself provides the main disadvantage of this method, which is the extensive number of computer runs required. One only need visualize extending the two-factor case of Figure 3 to 10 or 20 factors to see the extremely large number of computer runs required for any reasonably designed grid.

2. Random Search—Random search involves making simulation runs corresponding to combinations of factor values which are chosen at random in some way. Before a random search can proceed, the overall region of exploration must be defined. This can be done by specifying the upper and lower limits, U_i^* and L_i^*, respectively, for each controllable factor X_i.

Although there are many variations to conducting a random search [6], there are two basic

methods. Assume that there are k controllable factors. The first method involves choosing a random number R_i uniformly on the interval (L_i^*, U_i^*) for each $i = 1, \ldots, k$. Then the simulation run corresponding to $(X_1, \ldots, X_k) = (R_1, \ldots, R_k)$ is made, and the process is repeated a number of times with the point giving the best response being chosen as the "optimum." (This method of random search is illustrated for the two-factor case in Figure 4.) The second basic method of random search involves random sampling of points from the grid given by a factorial design. Thus, for the two-factor case this second method would have computer runs made which correspond to randomly selected points from the grid given in Figure 3.

The major advantage of random search is that there is, in essence, no requirement on the number of computer runs as there is for the application of a factorial design. As few (or as many) runs as desired may be made. Of course, among the disadvantages of random search is the fact that the search results in neither a systematic nor a thorough exploration of the response surface unless an extremely large number of computer runs are made.

B. Adaptive Search Techniques

It seems reasonable to assume that the results of previous computer runs may, indeed, provide valuable information which can be used in the search for an "optimum." For example, if any computer runs produce relatively bad responses, it may be well to avoid searching the region near the corresponding points. On the other hand, if a relatively good response is obtained at a point, it may be well to explore the region around that point more thoroughly. The two search techniques which serve as a basis for most adaptive methods are the single-factor method and Response Surface Methodology (RSM).

1. Single-Factor Method—The single-factor method [12], as its name implies, concentrates on one factor at a time. Although there are variations to the way in which this method is applied, only one procedure will be discussed here. Specifically, when there are k controllable factors,* $X_1, \ldots, X_k$, a starting point $(C_1, \ldots, C_k)$ is chosen, and a computer run made at this point. A step size Δ_1, is chosen, and the X_1 value is changed to $C_1 - \Delta_1$ and all the other X_i's remain at their starting values. A computer run is made at this new point. Similarly, a computer run is made at a point where $X_1 = C_1 + \Delta_1$ and the other X_i's remain the same. If the response at either of these two points is better than at the starting point, the search is extended in the direction of the most improvement, and with all the other factors held constant, a new computer run is made, corresponding to a value of $C_1 - 2\Delta_1$ or $C_1 + 2\Delta_1$ for X_1, as is appropriate. The search continues in the same direction as long as improvement continues to be made.

If a point is reached at which improvement is no longer being made, the best previous point or a point estimated by fitting a polynomial to the previously observed points is regarded as a new starting point, a step size Δ_2 is chosen, and the process is repeated, changing only X_2 while keeping the values of the other factors fixed. (After factor X_k is examined, the cycle is completed by then considering X_1 again.) This procedure continues with each factor, in turn, until either the available number of computer runs are exhausted or an "optimum" point is identified.

Figure 5 indicates the application of the single-factor method for two controllable factors X_1 and X_2. By referring to this figure, one can see that improvement over the starting point was indicated in

*It is assumed that these factors are listed in order of their estimated importance, if such estimates exist.

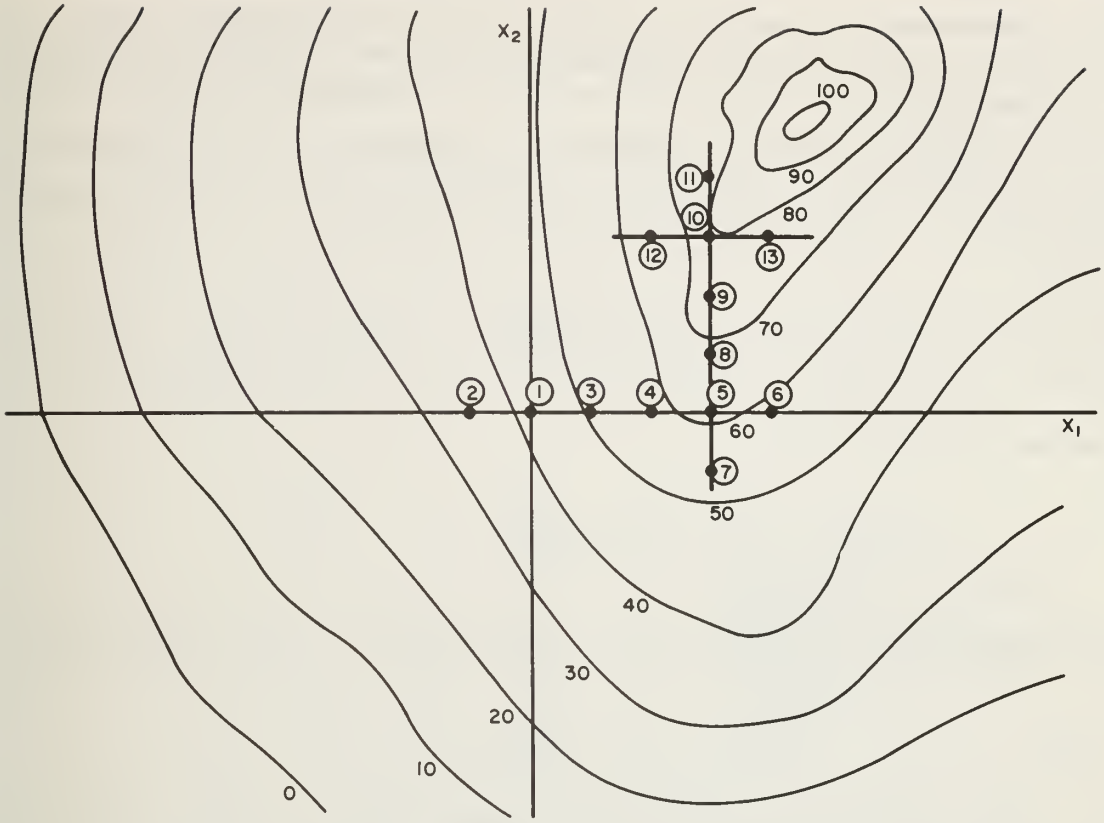


FIGURE 5. Application of the single-factor approach.

positive X_1 direction. Thus, points were taken in this direction. When point #6 was reached, the response was less than that at point #5, hence, an indication that the search should proceed from point #5 with X_1 held constant and X_2 varied.

Following the single-factor procedure outlined above, it is evident that in the case illustrated in Figure 5, point #10 will be identified as the "optimum". This is, of course, because the steps made in the X_1 direction and the X_2 direction provide points (#9, #11, #12, and #13) for which the response is lower than for point #10. This situation illustrates that although this adaptive method can make progress toward an optimum, it may provide a suboptimal point, not only because of a lack of computer runs, but also because it constrains search movement parallel to the axes.

2. *Response Surface Methodology*—The basis of Response Surface Methodology (RSM) is contained in a paper by Box and Wilson [3]. RSM permits the controlled factors to be varied simultaneously, so that there are no constraints on the directions in which a search must be conducted. Instead, a search may be in any direction. Essentially, RSM makes use of the assumption that the response surface can, in any local region, be approximated to a good degree by a hyperplane. That is, ignoring statistical variation, a good approximation to the response surface in any locality is given by the equation of a hyperplane:

$$\alpha_0 + \sum_{1}^{k} \alpha_i X_i.$$

If an initial experimental region (i.e., a locality) is defined by the boundaries $L_i \leq X_i \leq U_i$, it is often convenient to code the largest value of each factor as $+1$ and the smallest value of each factor as -1. For example, if factor X_j were to be investigated in the interval $20 \leq X_j \leq 60$, the coding would be given by $x_j = \frac{X_j - 40}{20}$, so that $x_j = -1$ is equivalent to $X_j = 20$, and $x_j = +1$ is equivalent to $X_j = 60$. If the factors $X_1, \ldots, X_k$ are transformed into the coded factors $x_1, \ldots, x_k$ in this manner, the equation

$$\alpha_0 + \sum_1^k \alpha_i X_i$$

is transformed into the equation

$$\beta_0 + \sum_1^k \beta_i x_i.$$

Thus, an estimate $\hat{y}$ of the value of the response y at the coded point $(x_1, \ldots, x_k)$ would be given by

$$\hat{y} = b_0 + \sum_1^k b_i x_i,$$

where b_i, obtained from an initial experiment by the method of least squares, is an estimate of β_i.

The estimates $(b_1, \ldots, b_k)$ determine the estimated gradient direction which is known as the path of steepest ascent. Computer runs corresponding to points on this path, which provides the approximate direction of predicted maximum response, are made until there is no improvement in the observed response, at which time the whole process may be repeated, usually within a smaller experimental region. When a near stationary region is located, additional experiments may be conducted to determine the approximate curvature of the response surface, and, thus, to a good degree of approximation, also determine the optimum point in this region. It is reasonable to assume, however, that this last step will rarely be used in dealing with practical simulation problems, because of the large number of computer runs required. Thus, the first phases of RSM (the initial experiment and the determination of the path of steepest ascent) will be of main interest in the current study.

In general, the initial experiment is based on a first order design. First order designs are experimental designs which have been developed specifically for fitting a hyperplane, i.e., a polynomial of first degree. Often used first order designs are the 2^k factorials and the 2^{k-p} fractional factorials [9], which are experiments involving only two values (or levels) of each factor $X_1, \ldots, X_k$, the Plackett-Burman designs [19], and the simplex designs [2]. The Plackett-Burman designs, like the 2^{k-p} fractional factorials, involve only two levels of each factor. However, they tend to provide initial experiments which require fewer computer runs, although they are more difficult to construct. The simplex designs result in initial experiments involving $k+1$ computer runs.

As may be noted by the reader, when k factors are being investigated, a minimum of $k+1$ points is needed to calculate the path of steepest ascent, i.e., to fit a hyperplane. However, when statistical variation exists, it may be desirable to use more than this minimum number to provide some estimate of

the size of the statistical variation. In some situations, on the other hand, it may be that best results are obtained by proceeding with the minimum possible number of runs.*

Figure 6 indicates the application of RSM for a two-factor situation. Here a 2^2 factorial, i.e., an experiment involving all four combinations of the two levels of both X_1 and X_2, was used to determine the path of steepest ascent, indicated in the figure. This path was followed until the observed response fell off, at which time a new 2^2 factorial in a smaller region was used to calculate a new path of steepest ascent. This process was repeated again, at which time the path crossed over the true optimum. If another 2^2 factorial were used in this locality, a near stationary region would be indicated. Thus, if enough computer runs were available, the curvature of the response surface should be investigated in this locality. Applicable designs for this purpose are second order designs which are discussed in Cochran and Cox [9].

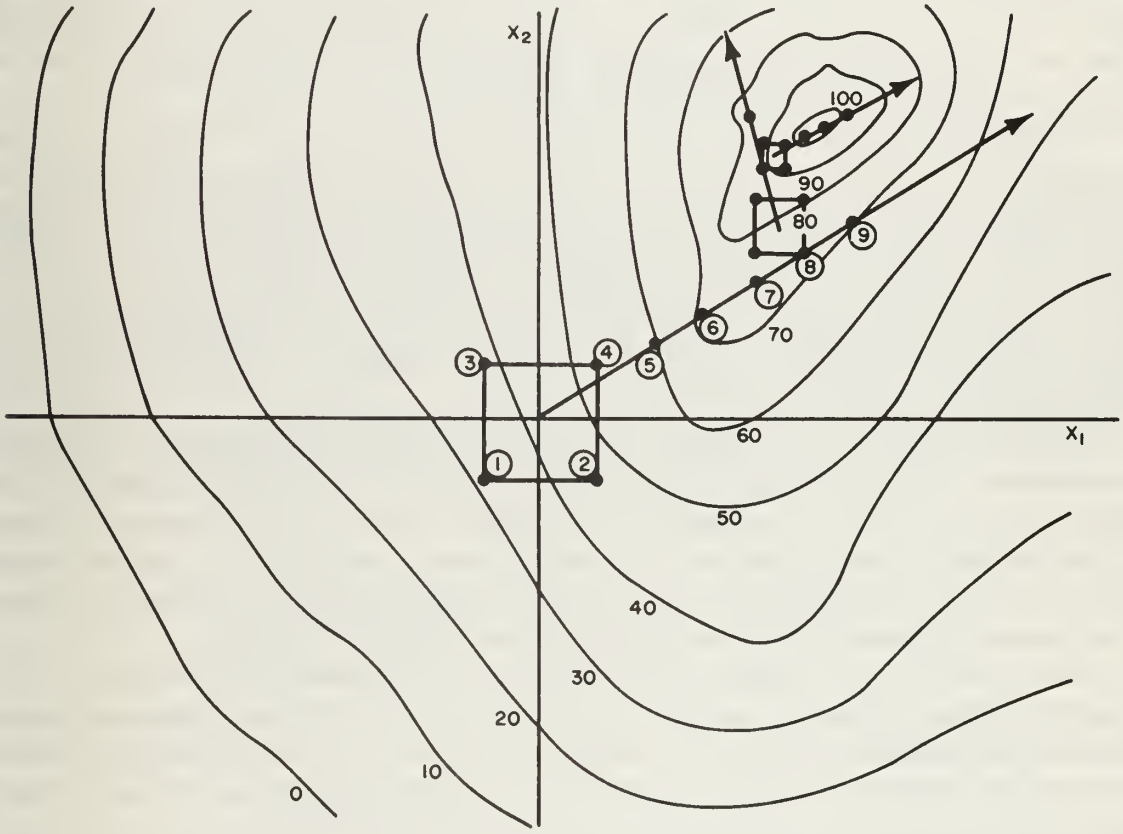


FIGURE 6. Application of RSM.

When compared with the single-factor method, RSM is seen to have the advantage of providing a search which is guided in the most likely direction for improvement. However, the requirement of an initial experiment by RSM means that a considerable number of computer runs (at least $k + 1$) are spent in a specific locality before any searching is actually done, while a search based on the single-factor method may be able to move away from the starting point after a minimum of three runs.

*Brooks and Mickey [7] reach this conclusion based on a criterion pertaining to the size of the angle between the true gradient and the estimated gradient.

IV. SOME DRAWBACKS

The use of external search techniques in a simulation situation is hindered by two characteristics of computer simulation:

(1) An excessive number of controllable factors.

(2) The presence of statistical variation.

Although search techniques such as those described in section III may appear to contain well-defined procedures, it must be realized that unless they are modified to take into account these complications, erroneous conclusions may result.

Another way of viewing the complication of too many factors is that there is an extremely limited number of computer runs. This is easily seen for the case of a factorial design. If, for example, there are 50 controllable factors to be investigated by means of a factorial design involving only two different values of each factor, the required number of computer runs will be $(2)^{50}$, that is, over 1,126,000,000,000,000,000 computer runs, truly an astronomical requirement. It can readily be seen, therefore, that for most practical problems, a factorial design cannot be used.

Although not as staggering as the requirements for a factorial design, the number of computer runs required by the other techniques also tend to be excessive. For example, taking the same case, it is reasonable to assume that, at a minimum, in the neighborhood of 200–1000 computer runs will be required to make only one complete cycle through the single-factor method. Likewise, for RSM, a minimum of 51 computer runs would be required just to establish the path of steepest ascent, to say nothing about following the path and perhaps repeating the process. Of course, random search does not require any minimum number of computer runs. However, it is doubtful that much success would result unless considerably more than 50 computer runs were made.

A second complication arises because of the probabilistic nature of the observed response which, as mentioned in section II, is usually contaminated with statistical error (statistical variation), and thus may be far from the actual response surface. For example, suppose the observed response for a given computer simulation is the time required for the performance of a certain task. This time may actually vary about an expected value (or mean) of 60 minutes, but because of statistical variation, one computer run may produce an observed response as low as 30 minutes or as high as 90 minutes.

In the extreme case, the response of interest may be the probability of some event, e.g., the probability of killing an enemy submarine. In simulations such as this where the true responses are probabilities, the observed responses are either 1 (kill) or 0 (no kill). Thus, the observed response produced by one computer run contributes little or no help in the search for an optimum, because it can only be zero or a one.

Despite their drawbacks, external methods do offer potential for "optimization" in simulation. If these methods could be automated, at least to a certain degree, efficient application of these methods in the "optimization" process would be expedited. The possibility of automation is explored in the next section.

V. THE POTENTIAL USEFULNESS AND DESIGN REQUIREMENTS OF AN "OPTIMIZER" PROGRAM

Because search techniques tend to provide the most useful means of "optimization," the existence of these techniques in an "Optimizer" computer program should prove of value to the computer simulation user who desires an "optimal" solution to his problem. Such an "Optimizer" program could

be constructed independently of any specific computer simulation, and would thus be general enough to provide extreme flexibility and, therefore, wide applicability. In addition, the user would not need an extensive knowledge of the mathematics underlying the techniques, and would not have to contend directly with overcoming the drawbacks of applying them, since this could be handled within the structure of the "Optimizer." The "Optimizer" would have to be designed in such a manner that it could be granted access to the inputs and outputs of the computer simulation. In addition, the design would have to allow the computer simulation user to supply pertinent additional data (such as the starting region for the search and the number of computer runs that are available).

Figure 7 summarizes the design concept of a general "Optimizer" program which could be tailored to any specific computer simulation situation by additional information supplied as input by the user. This "Optimizer," as discussed here, would be designed under the three assumptions made in section II.B. These are:

1. A single response is under investigation.
2. The controllable factors are quantitative and continuous (at least approximately).
3. The uncontrollable factors have known values or generate random variables with known probability distributions.

In essence, the "Optimizer" would be obviating the need for an "expert" who is familiar with the various search techniques and the underlying mathematics, because they would be contained within the "Optimizer" itself. As indicated in Figure 7, the simulation user would be able to communicate to the "Optimizer" information in addition to the standard simulation input. This information would be used by the "Optimizer" to determine the best search technique (or techniques) to be adopted for the specific simulation situation. Thus, the "Optimizer" as visualized would have a double optimization task. Not only would it attempt to find an "optimum" solution, but it would also attempt to determine the optimum "optimizing" technique(s) to be used in any situation.

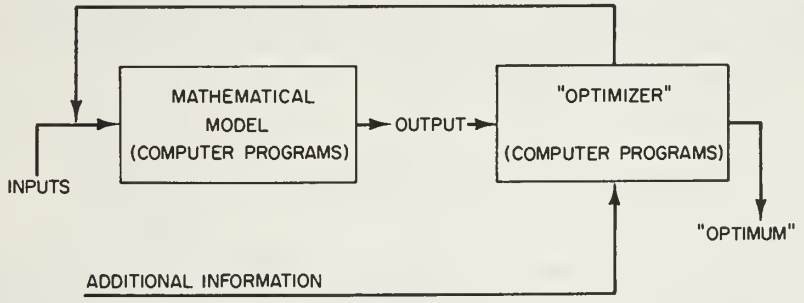


FIGURE 7. Ultimate objective of research.

An "Optimizer" should, of course, be primarily designed to provide ample opportunity to locate and explore a region of improved response, if such a region exists. In view of the complications discussed in section IV, the "Optimizer" must also be designed to take into account the presence of statistical variation and the large number of factors usually under investigation. Thus, an "Optimizer" should be designed to afford reduction of statistical variation to help prevent erroneous conclusions and to provide for necessary screening (sensitivity analysis) to eliminate factors of marginal importance.

Unfortunately, satisfying each of these specifications requires a number of computer runs, so that

trade-offs must be made. For example, in allocating computer runs, one must decide between estimating the true response at a relatively small number of points with a high degree of confidence, i.e., small variance, or at a larger number of points with a lesser degree of confidence, i.e., large variance. At this point, the three design requirements will be examined briefly.

A. Reduction of Statistical Variation

The main reason for reducing for controlling the statistical variation is to try to insure that the "optimum" point identified does truly produce a good solution. As a simple example, consider a computer simulation which produces responses which arise from Normal distributions. Suppose that ten computer runs (corresponding to different points selected at random) are made, where points #1 through #9, inclusive, each generate, independently, an observed response which has mean (expected value) μ and variance σ^2, while point #10 generates an observed response which has mean $\mu+\sigma$ and variance σ^2. Because point #10 is the best point, from the viewpoint of having the largest expected value, one would hope that the search of these ten points would result in a choice of point #10 as the "optimum." The probability of this choice being made can easily be calculated.

If y_i denotes the observed response at point i, for $i=1, \ldots, 10$, then

$$P(\text{Point \#10 is chosen}) = P(y_i < y_{10} \quad \text{for } i=1, \ldots, 9)$$

$$= \prod_{i=1}^{9} P(y_i < y_{10})$$

$$= \prod_{i=1}^{9} P(0 < y_{10} - y_i).$$

By standardizing the random variable $y_{10}-y_i$, this probability may be expressed in terms of a random variable Z having a Normal distribution with mean 0 and variance 1, which is a distribution well-tabulated in most statistics books. It follows that

$$P(\text{Point \#10 is chosen}) = \left[P\left(Z > \frac{-\sqrt{2}}{2}\right)\right]^9 = 0.08.$$

Thus, there is only an 8 percent chance of choosing as the "optimum," the point which is actually the best of the ten considered. Because of the statistical variation there is a large chance (92 percent) of making an incorrect choice of the "optimum."

If, instead of taking only 1 observation (i.e., one computer run) at each point, 12 observations were taken at each point and the average used as the observed response, it can be verified that there would be only a 6-percent chance of making an error in the selection of the "optimum." It should be noted, however, that this gain in accuracy is offset by the fact that the 120 computer runs used to reduce the variation at 10 points could have been used instead to explore the observed responses at an additional 110 points. If all of these 110 points produced true responses (i.e., expected values) less than or equal to μ, quite a bit would have been gained by not examining them. However, if only a few (or even one) of these points had a true response quite a bit greater than $\mu+\sigma$, the additional accuracy obtained at the original 10 points would have precluded the possible identification of any of these points as "optimum."

For example, if only 1 of the additional 110 points, say point #110, produced an expected value of $\mu + 6\sigma$, while the others produced expected values on the interval $(\mu, \mu + \sigma)$, it can be easily verified that there is at least a 98-percent probability that point #110 would indeed be identified as the "optimum." However, because the responses (observed and true) corresponding to unexplored points are never known, it is often difficult to decide how many computer runs to devote to reducing the statistical variation.

Up to this point only one method for reducing the amount of variation associated with an observed response has been discussed. This method involves the use of an average observed response based on a number of observations at the same point, that is, a number of repeated computer runs under the same combination of values of input factors. The main drawback to doing this is that computer runs are rapidly used up. Another method is to use search techniques which have internal replication incorporated within them. An example will illustrate the concept of internal replication.

Consider Figure 8, which depicts two techniques for estimating the change in response due to increasing X_1 or increasing X_2. In technique I, two repeated observations are taken at each of three points. It is easily seen that the change in response occurring from an increase in X_1 may be estimated by subtracting the average observed response at point #1 from the average observed response at point #2, and similarly for X_2. Thus, the estimate of change in response due to an increase in X_1 is based on a total of four observations, as is the estimate of change in response due to an increase in X_2. In technique II, which is a 2^2 factorial design, a single observation is taken at each of four points. An estimate of change in response occurring from an increase in X_1 may be obtained by subtracting the average observed response at points #1 and #3 from the average observed response at points #2 and #4. Likewise, an analogous estimate holds for the effect of X_2.

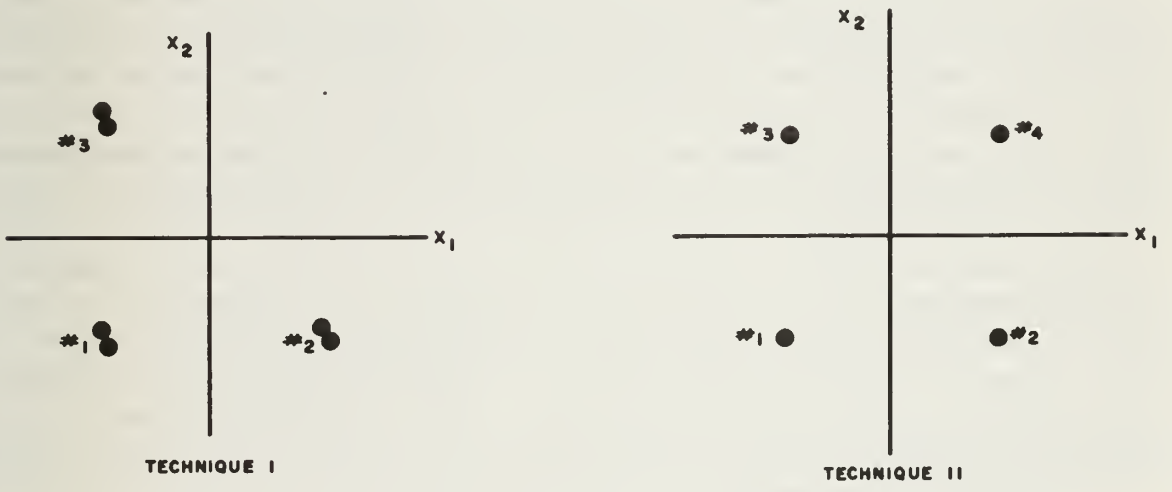


FIGURE 8. Internal replication.

In technique II, then, as in technique I, the estimate of change in response due to an increase in X_1 is based on a total of four observations, as is the estimate of change in response due to an increase in X_2. Thus, as far as measurement of the effects of an increase in X_1 or an increase in X_2, the two techniques produce estimates which are based on the same effective number of observations, and hence which have the same amount of statistical variation associated with them. Because techniques I requires a total of six observations while technique II requires only four observations, technique II would be

preferred to technique I for obtaining these estimates. As an aside, it should be noted that technique I provides better estimates of the true responses at points #1, #2, and #3 than does technique II, while technique II yields an estimate of the true response at point #4, which technique I does not.

As a general guideline, the smaller the differences between true responses at "optimum" and "nonoptimum" points, relative to the amount of statistical variation, the more reduction of variation is required. However, unless estimates of the differences in true response and the amount of statistical variation exist, this guideline proves essentially useless. If no a priori estimates are available, the "Optimizer" will, hopefully, be able to develop these estimates in order that a reasonable decision concerning the reduction of statistical variation can be made. As can be seen from this brief discussion, the decision of how much reduction of variation is required and how it should be effected is a difficult one to make, and does pose a serious problem in the design of the "Optimizer."

B. Screening of Factors (Sensitivity Analysis)

Because of the usual constraints of a large number of factors and a limited number of computer runs, it is often impossible to investigate thoroughly all factors under consideration. What is required, therefore, is some means of making the available number of computer runs and the number of factors compatible. Because there are usually some factors under initial consideration which actually have little or no effect on the response, an attempt to identify and eliminate these factors at an early stage in "optimization" by means of some screening process should improve the compatibility. In general, there are two types of screening processes: (a) direct elimination of factors, and (b) grouping of factors.

1. Direct Elimination of Factors—Suppose that in a certain simulation there are K factors under consideration, but because of a limited number of computer runs, it is desired to investigate only k factors where $k < K$. In some cases the existence of prior knowledge, based on simulation of similar problems or on considerations of some of the processes being simulated, may allow an educated guess of the relative important of at least some of the factors. For example, because of previous experience, the user of a certain ASW simulation may be quite certain that submarine depth will have little or no effect on the response when compared with the rest of the factors. In this situation, then, submarine depth could be eliminated from the investigation by keeping this factor fixed at a constant value throughout the search for an "optimum."

Unfortunately, because prior knowledge such as this often exists only vaguely as intuition, if it exists at all, elimination of factors in this manner will probably not be too useful. Of course, in the absence of prior knowledge, factors may be eliminated by random selection. Thus, if only k factors are to be investigated, k of the original K factors may be selected at random for inclusion in the investigation, with the remaining $(K-k)$ factors eliminated from further consideration. Needless to say, not much merit can be found in such a procedure, although in extreme cases (i.e., when no other approach seems feasible) it may be the only alternative.

Instead of attempting to eliminate factors before any computer runs are made, it may sometimes prove of value to expend a portion of the available computer runs in a preliminary search having as one of its goals the reduction in the number of factors to be considered. Of necessity, such a preliminary search would, as a rule, involve considerably less computer runs than factors, thus giving rise to confounded estimates, that is, estimates which are "mixed together" and impossible to separate by statistical analysis. Confounded estimates provide ambiguous results which may, if interpreted incorrectly, lead to completely erroneous conclusions about the location of an "optimum" point.

Because unconfounded estimates of the effects of K factors cannot be obtained without a minimum of $K+1$ computer runs, the confounding problem will present severe drawbacks to the usefulness of preliminary search procedures. Nonetheless, they may still prove of enough value to be included in the "Optimizer." The most likely candidates for preliminary search procedures are the random balance designs described by Satterthwaite [21], Budne [8], and Anscombe [1], although reported practical experience with these techniques has been extremely limited.

2. Grouping of Factors—If factors are grouped together in some manner and the group of factors is considered as a single factor, much economy in the number of computer runs is possible. Such grouping may be afforded by knowledge of the simulation situation. For example, in a simulation involving m destroyers, the effect of changes in destroyer speed on the response may be of interest. Thus, m factors $X_1, \ldots, X_m$ are under consideration, where X_i corresponds to the speed of the ith destroyer. In most situations, however, it can be assumed that each of these factors has an equivalent effect on the response. That is, an increase (or decrease) in the value of factor X_i has essentially the same effect on the response as the same increase (or decrease) in the value of factor X_j. In view of this, there is no reason why computer runs should be used to run combinations of increases and decreases in the values of $X_1, \ldots, X_m$. Instead, the values of all the factors $X_1, \ldots, X_m$ could be increased (or decreased) simultaneously. Thus, in essence, the m factors $X_1, \ldots, X_m$ would become equivalent to one grouped factor, because whenever the value of factor X_1 was increased or decreased, so would the values of the remaining $m-1$ factors $X_2, \ldots, X_m$. If two values of each of these factors are involved, the grouping would eliminate 2^m-2 computer runs.

Another means of grouping factors is for the user to reduce a definite number of factors to a smaller number by consideration of underlying mathematical and physical relationships between the factors. For example, instead of using the factors "velocity" and "time" in a given simulation, the user has the option of using the factor "distance." Likewise, instead of "length," "width," and "height," the user may decide to use the factor "volume." Such a procedure is somewhat related to a discipline known as dimensional analysis, which may be used to group physical parameters into convenient combinations that reduce the number of factors under consideration. Dimensional analysis is based on the fact that the quantities on both sides of an equation must have the same dimensions. Because of this fact, it is often possible to arrive at the relationships within a group of factors without detailed theory. These relationships may then be used to define all factors in the group in terms of a selected subset of these factors. The reader interested in dimensional analysis is referred to Bridgeman [4] or Langhaar [17].

In the absence of any assumptions or knowledge of how the various factors are related, a grouping of factors may simply be made on a random basis. Although such a grouping would not be expected to be as efficient as a grouping based on prior knowledge, it may in certain cases provide a relatively good payoff. Grouping of factors has been discussed, under various assumptions, by Watson [23], Patel [18], and Gurnow [13]. Unfortunately, practical applications appear to be lacking.

C. Location and Exploration of Region of Improved Response

The primary objective of an "optimization" technique is to locate an "optimum" which, in fact, does represent an improved solution. In order for this objective to be met, any technique or combination of techniques must have a good chance of locating a region (if it exists) in which the responses are larger than in the initial search region. Such techniques can, for the most part, be based on the factorial design, random research, single-factor approach, and RSM, which have been previously discussed.

However, to deal with the large number of factors and the statistical variation usually present in the simulation situation, modifications to these techniques may prove valuable, especially in early stages of the "optimization" process. Some possible modifications have been mentioned in this section. Their application will, to a large degree, be governed by the characteristics of the simulation and the amount of prior knowledge about the simulation which the user possesses.

VI. SUMMARY

"Optimization" in computer simulation will not simply involve the choice of a search technique. Instead, it will involve the choice of an "optimization" strategy consisting of a search technique, a screening method, and possible modifications for reducing the amount of statistical variation present. At this juncture it seems feasible to determine the most efficient strategies and automate their use in an "Optimizer", which should prove a valuable alternative to common trial-and-error procedures.

Work in this direction is currently in progress.

REFERENCES

[1] Anscombe, F. J. (1959). "Quick Analysis Methods for Random Balance Screening Experiments," Technometrics, *1*, p. 195.

[2] Box, G. E. P. (1952). "Multi-factor Designs of First Order," Biometrika, *39*, p. 49.

[3] Box, G. E. P., and Wilson, K. B. (1951). "On the Experimental Attainment of Optimum Conditions," Journal of the Royal Statistical Society (Series B), *13*, p. 1.

[4] Bridgeman, P. (1931). *Dimensional Analysis*, Yale University Press, New Haven, Connecticut.

[5] Brooks, S. H. (1959). "A Comparison of Maximum—Seeking Methods," Operations Research, *7*, No. 4, p. 430.

[6] Brooks, S. H. (1958). "A Discussion of Random Methods for Seeking Maxima," Operations Research, *6*, p. 244.

[7] Brooks, S. H. and Mickey, M. R. (1961). "Optimum Estimation of Gradient Direction in Steepest Ascent Experiments," Biometrics, *17*, p. 48.

[8] Budne, T. A. (1959). "The Application of Random Balance Designs," Technometrics, *1*, p. 139.

[9] Cochran, W. G., and Cox, G. M. (1957). *Experimental Designs* (2nd Edition), John Wiley and Sons, Inc., New York.

[10] Davies, O. L., ed. (1967). *Design and Analysis of Industrial Experiments*, Hafner Publishing Co., New York.

[11] Flood, M. M., and Leon, A. (1965). "A Universal Adaptive Code for Optimization (GROPE)," in *Recent Advances in Optimization Techniques*, Vogl, T. and Lavi, A., eds., John Wiley and Sons, Inc., New York.

[12] Friedman, M., and Savage, L. J. (1947). Chapter 13, *Techniques of Statistical Analysis*, Eisenhart, C., Hastay, M., and Wallis, W., McGraw-Hill Book Co., New York.

[13] Gurnow, R. N. (1965). "A Note on G. S. Watson's Paper," Technometrics, *7*, p. 444.

[14] Hooke, R. and Jeeves, T. A. (1961). "Direct Search Solution of Numerical and Statistical Problems," Journal of the Association for Computing Machinery, p. 212.

[15] Jacoby, J. E., and Harrison, S. (1962). "Multi-Variable Experimentation and Simulation Models," Nav. Res. Log. Quart., *9*, No. 2 p. 121.

[16] Koopman, B. O., and Berger, H. M. (1967). "Use and Misuse of Simulations," in *Digital Simulation in Operations Research*, Hollingdale, S. H., ed., American Elsevier Publishing Co., New York.

[17] Langhaar, H. L. (1952). *Dimensional Analysis and Theory of Models*, John Wiley and Sons, Inc., New York.

[18] Patel, M. S. (1962). "Group Screening with More Than Two Stages," Technometrics, *4*, p. 209.

[19] Plackett, R. L., and Burman, J. P. (1946). "The Design of Optimum Multifactor Experiments," Biometrika, *33*, p. 305.

[20] Rosenbrock, H. H. (1960). "Automatic Method for Finding the Greatest or Least Value of a Function," The Computer Journal, *3*, p. 175.

[21] Satterthwaite, F. E. (1959). "Randon Balance Experimentation," Technometrics, *1*, p. 111.

[22] Smith, D. E. (1969). "Optimization Within Computer Simulation: An Application of Response Surface Methodology," 24th Military Operations Research Symposium.

[23] Smith, D. E. (1970). "An 'Optimizer' for Use in Computer Simulation: Background and Design Concepts," Report No. 4352.11–R–1, HRB-Singer, Inc., State College, Pa.

[24] Watson, G. S. (1961). "A Study of the Group Screening Method," Technometrics, *3*, p. 371.

A COST ANALYSIS OF SAMPLING INSPECTION UNDER MILITARY STANDARD 105D

Gerald G. Brown

California State University, Fullerton

and

Herbert C. Rutemiller

California State University, Fullerton

ABSTRACT

Military Standard 105D has been almost universally adopted by government and private consumers for the lot-by-lot sampling inspection of product which may be inspected on a dichotomous basis.

The plan specifies, for each lot size, a random sample size and set of acceptance numbers (maximum allowable number of defectives in each sample). The acceptance numbers are based upon the binomial distribution and depend upon the quality required by the purchaser. Where several consecutive lots are submitted, a shift to less severe ("reduced") inspection or more severe ("tightened") inspection is specified when the ongoing quality is very high or low. Further experience permits a return to normal sampling from either of these states.

This paper examines the long range costs of such a sampling scheme. The three inspection types are considered as three distinct Markov chains, with periodic transitions from chain to chain. The expected sample size and the expected proportion of rejected product are determined as a function of the two parameters under control of the manufacturer, lot size and product quality. Some numerical examples are given which illustrate how to compute the overall cost of sampling inspection. Suggestions are made concerning the choice of parameters to minimize this cost.

1. INTRODUCTION

One of the major fields of statistical quality control is acceptance sampling. An increasing number of private and government purchasers of manufactured products require the sampling inspection of each shipment, or lot, and will accept the lot only if a quality standard is passed by the sample. Rejected lots may be returned to the manufacturer, purchased with a price concession, subjected to 100 percent screening, or scrapped. Clearly, there are costs involved for both the sampling inspection and the disposal of rejected lots. In this paper, we examine the cost aspects of sampling inspection according to the most widely used plan in the United States—the attributes sampling scheme designated Military Standard–105D [6].

In attributes sampling plans, a random sample of n individual items is selected from a lot, and each item is classified simply as acceptable or unacceptable. The lot is accepted if c or fewer items out of n are unacceptable. The conditional probability of acceptance, P_a, given that a proportion, p, of the lot is unacceptable, is given approximately by the cumulative binomial probability.

$$P_a = \sum_{y=0}^{c} \binom{n}{y} p^y (1-p)^{n-y}. \tag{1}$$

This assumes that the sample size is small relative to the lot size.

The long-run cost contribution of such a sampling plan may be computed as a function of overall product quality, p, as follows:

Let

C = cost per manufactured item as a result of sampling inspection;

C_1 = cost of inspecting a single item;

C_2 = cost per rejected item for the disposal of rejected lots, e.g., for 100 percent inspection of all items in the lot;

f = fraction of product inspected as sample;

q = fraction of product rejected;

N = size of each lot submitted;

then

$$C = C_1 f + C_2 q;$$

$$E[C] = C_1 n/N + C_2 \left(\frac{N-n}{N}\right)(1 - P_a).$$

For example, lots of size 500 from a process producing 5-percent defectives are submitted to sampling inspection according to the plan $n = 20$, $c = 2$. The cost of sampling inspection is \$2.00 per item and rejected lots must be 100 percent screened also at a cost of \$2.00 per item. By using Equation (1), binomial probability tables [9] give $P_a = 0.925$ and

$$E[C] = 2.00(20/500) + 2.00\left(\frac{480}{500}\right)(0.075)$$

$$= 0.08 + 0.144$$

$$= \$0.224.$$

This is the expected cost per manufactured item as a consequence of sampling inspection.

In practice, most attribute sampling plans have somewhat more complicated rules for their application. Dodge and Romig [1] have devised a set of attribute plans based upon minimum cost, assuming a desired incoming quality, with specific requirements for consumer protection against very bad lots. Hald [3] has greatly expanded this idea, and developed sampling plans which minimize cost for any prior distribution. However, today by far the most popular set of plans is that published by the Department of Defense in 1963, known as MIL–STD–105D. It is estimated that more than 13,000 industrial plants in the western world are currently involved with this standard, either as vendors or purchasers.

The sampling plans in MIL–STD–105D are not based upon the minimum-cost concepts of Hald. The plans are indexed by lot size and by a number designated "Acceptable Quality Level." The *AQL* is always specified by the purchaser, and is defined as the value of p which will lead to a high conditional probability of acceptance, P_a. The precise value of P_a corresponding to the *AQL* varies with lot size and *AQL*. The domain for P_a is about 0.89–0.99.

TABLE 1. *Some representative sample sizes and acceptance numbers (shown in body of table) from MIL–STD–105D, Table II–A, for various lot sizes and* AQL *percent defective. The arrow signifies to use the plan on the row at the end of the arrow. Plans shown are for general inspection level II in MIL–STD–105D.*

Lot size	Sample size	*AQL*							
		0.40	0.65	1.0	1.5	2.5	4.0	6.5	10.0
2–8	2	↓	↓	↓	↓	↓	↓	0	↓
9–15	3	↓	↓	↓	↓	↓	0	↑	↓
16–25	5	↓	↓	↓	↓	0	↑	↓	1
26–50	8	↓	↓	↓	0	↑	↓	1	2
51–90	13	↓	↓	0	↑	↓	1	2	3
91–150	20	↓	0	↑	↓	1	2	3	5
151–280	32	0	↑	↓	1	2	3	5	7
281–500	50	↑	↓	1	2	3	5	7	10
501–1,200	80	↓	1	2	3	5	7	10	14
1,201–3,200	125	1	2	3	5	7	10	14	21
3,201–10,000	200	2	3	5	7	10	14	21	↑
10,001–35,000	315	3	5	7	10	14	21	↑	↑
35,001–150,000	500	5	7	10	14	21	↑	↑	↑
150,001–500,000	800	7	10	14	21	↑	↑	↑	↑
500,001 up	1,250	10	14	21	↑	↑	↑	↑	↑

Table 1 shows some typical plans for normal sampling inspection from MIL–STD–105D. Once a sampling plan has been selected, the user is required to keep a historical record of lot-by-lot experience. Criteria are presented in MIL–STD–105D for an alteration of the values of n and c when the experience over several lots shows either unusually good or unusually bad quality. The rules are as follows [6]:

1. A switch from the normal values of n and c to "reduced" inspection is permissible when:

a. Ten consecutive lots have been accepted.

b. The total number of defectives in the 10 lots does not exceed a critical value supplied in Table VIII of MIL–STD–105D. Some examples of these critical values are shown in Table 2.

c. Production is continuous.

d. Reduced inspection is considered desirable by the responsible authority.

Under reduced inspection, n is substantially decreased. Two numbers, c and $r(>c)$ are supplied. Lots are accepted if the number of defectives is less than r. However, if a lot has more than c defectives, normal inspection must be resumed on the next lot.

2. A switch to "tightened" from normal inspection is required when two of the most recent five lots have been rejected. Under tightened inspection, n is the same as for normal inspection, but c is reduced. A return to normal is permitted when five consecutive lots have been accepted. If tightened inspection is still in use for 10 consecutive lots, however, sampling inspection must be discontinued entirely.

Hence, in attributes sampling by this scheme, the long run costs will be dependent on the proportion of lots inspected under normal, reduced, and tightened sampling plans. Koyama [4] has examined the severity of the switching rules in MIL–STD–105D, using signal flow graph theory. Mendizabal [5] has performed some limited Monte Carlo studies in the same area. The Koyama results indicate that, for lots with consistent *AQL* quality, the switching rules are severe, with the probability of entering tightened inspection relatively high.

TABLE 2. *Some limit numbers from MIL–STD–105D, Table VIII for transfer from normal to reduced inspection. Numbers in body of table are maximum number of defectives in most recent 10 lots.*

Number of items in last 10 lots	*AQL*							
	0.40	0.65	1.0	1.5	2.5	4.0	6.5	10.0
20–29								0
30–49							0	0
50–79						0	0	2
80–129					0	0	2	4
130–199				0	0	2	4	7
200–319			0	0	2	4	8	14
320–499		0	0	1	4	8	14	24
500–799	0	0	2	3	7	14	25	40
800–1,249	0	2	4	7	14	24	42	68
1,250–1,999	2	4	7	13	24	40	69	110
2,000–3,149	4	8	14	22	40	68	115	181
3,150–4,999	8	14	24	38	67	111	186	
5,000–7,999	14	25	40	63	110	181		
8,000–12,499	24	42	68	105	181			
12,500–19,999	40	69	110	169				
20,000–31,499	68	115	181					
31,500–49,999	111	186						
50,000 up	181	301						

In the present paper, we develop, for MIL–STD–105D an expression for expected inspection cost as a function of process quality, p, and lot size, N, the two parameters which may be controlled, to some extent, by the manufacturer. We consider on-going inspection as a dynamic process, where recycling among normal, reduced, and tightened inspection will occur as a large number of lots is submitted. The solution produces the expected proportion of lots on normal, reduced, and tightened inspection, plus the expected proportion of rejected lots, for each (AQL, p, N) combination. With appropriate assumptions on the disposal of rejected lots and the cost of inspection, the expected cost of a MIL–STD–105D plan may then be obtained.

2. THEORY

We make two assumptions throughout this section. First, it is assumed that rejected lots incur a known cost per item for every item in the lot as a consequence of 100 percent screening. The second assumption concerns the re-institution of sampling inspection after it has been discontinued. MIL–STD–105D, section 8.4, states "In the event that 10 consecutive lots or batches remain on tightened inspection (or such other number as may be specified by the responsible authority), inspection under the provisions of this document should be discontinued pending action to improve the quality of submitted material." We assume that this cessation is in force for the next 10 lots, which will be screened 100 percent; normal sampling inspection will be resumed after this screening period.

The following additional notation will be needed in this section:

N = lot size;

n = sample size under normal inspection;

n_T = sample size under tightened inspection;

n_R = sample size under reduced inspection;

$P_{A,N}$ = probability of accepting a lot in normal inspection;

r^* = maximum number of defectives in 10 accepted lots under normal inspection to qualify for reduced inspection (Table VIII, MIL–STD–105D);

$P'_{A,N}$ = probability of meeting, at the first opportunity, the maximum defectives criterion of Table VIII for a transfer to reduced inspection, given that the most recent 10 lots are accepted;

$P_{A,T}$ = probability of accepting a lot in tightened inspection;

$P'_{A,R}$ = probability of accepting a lot in reduced inspection, but with a return to normal required;

$P_{A,R}$ = probability of accepting a lot in reduced inspection, with a continuation of reduced inspection;

X_N = number of adoptions of normal sampling during the inspection of L lots;

X_R = number of adoptions of reduced sampling during inspection of L lots;

X_T = number of adoptions of tightened sampling during inspection of L lots.

After L lots have been inspected under the MIL–STD–105D scheme, the total number of items in these lots will be $N \cdot L$. Suppose that l_n of these lots have been under normal inspection, l_r under reduced, and l_t under tightened. Then the fraction of product inspected by sampling will be

$$f = [l_R n_R + l_N n + l_T n_T]/NL. \tag{2}$$

Note that $l_N + l_R + l_T \leq L$, in general, since sampling inspection may be discontinued prior to the Lth lot.

The fraction of product subjected to 100-percent screening will be

$$E[q \mid l_N, l_T, l_R, X_R] = \left\{L - l_N P_{A,N} - l_T P_{A,T} - l_R + X_R \left[\frac{1 - P_{A,R} - P'_{A,R}}{1 - P_{A,R}}\right]\right\}/L. \tag{3}$$

Finally, the expected cost of sampling will be

$$E[C] = C_1 E[f] + C_2 E[q \mid l_N, l_T, l_R, X_R]. \tag{4}$$

Examining f and q, we note that these variables are functions of the parameters N, $P_{A,N}$, $P_{A,R}$, $P'_{A,R}$, $P_{A,T}$ and are linear functions of the variables l_N, l_R, l_T, X_R. It remains to find the numerical values for the parameters and the expectations of the four random variables.

Once a lot size, N, AQL, and inspection level have been specified, MIL–STD–105D provides both the sample size and acceptance number for normal, tightened, and reduced inspection. When product of quality p is supplied, the values of $P_{A,N}$, $P_{A,T}$, $P_{A,R}$, $P'_{A,R}$ may be computed directly from Equation (1) or other appropriate binomial sums. In practice, the Poisson approximation,

$$P_A = \sum_{y=0}^{c} (np)^y \exp(-np)/y!, \tag{5}$$

is usually employed, since p is small and n reasonably large. Molina [7] gives extensive tables of these Poisson probabilities. In the numerical examples which follow, Poisson probabilities were used in place of the binomial.

The computation of $P'_{A,N}$, the probability that a sequence of 10 accepted lots on normal inspection will meet the criterion of Table VIII in MIL–STD–105D for a transfer to reduced inspection, can best be shown through a numerical example. Suppose that an *AQL* of 1.5 and lot size of 400 is specified, and that material of *AQL* quality is submitted. The sampling plan is $n=50$, $c=2$; r^*, the limit for transfer to reduced inspection is 3, or fewer in 10 lots. Now, each of the 10 accepted lots contains two, one, or zero defectives. Hence, a truncated Poisson distribution is appropriate for the distribution of defectives in these lots:

c	Poisson probability	Conditional probability
0	0.4724	0.4923
1	0.3543	0.3692
2	0.1328	0.1385

The probabilities in column 3 were obtained by dividing the values in column 2 by their sum.

Next, $P'_{A,N}$ is the sum of the following multinomial probabilities:

Ten-sample Σc_i	Probabilities
3	$90(0.4923)^8(0.3692)(0.1385)+120(0.4923)^7(0.3692)^3$
2	$10(0.4923)^9(0.1385)+45(0.4923)^8(0.3692)^2$
1	$10(0.4923)^9(0.3692)$
0	$(0.4923)^{10}$

which gives, for this example, $P'_{A,N}=0.0890$.

For the general case, let

$$f(c_i), c_i=0, 1, \ldots, c \tag{6}$$

be the truncated Poisson probability distribution for lot i, accepted under normal inspection, and

$$g_k(s), s=0, 1, \ldots, kc$$

be the distribution for the total number of defectives in k accepted lots. Then the probability distribution for s is the convolution

$$g_{10}(s)=f(c_1)*f(c_2)*\ \ldots\ *f(c_{10});$$

$$\sum_{i=1}^{10} c_i = s;\ s=0, 1, \ldots, 10c;$$

while

$$P'_{A,N}=\sum_{s=0}^{r^*} g_{10}(s). \tag{7}$$

The calculation of $P'_{A,N}$ becomes lengthy, particularly for large lots and/or *AQL* values, where the number of partitions of Σc_i into 10 sample points is large. However, these cases may be handled with a normal approximation, as a consequence of the central limit theorem. If μ and σ are the parameters of a truncated Poisson distribution, then the sum of 10 random observations is approximately normally distributed with parameters 10μ and $\sqrt{10}\,\sigma$. We have tested this approximation and found it to be excellent for large values of the limit numbers in Table VIII. Even for the numerical example above, where c and r^* are 2 and 3, respectively, the normal approximation is reasonably good. The mean and standard deviation for the conditional Poisson distribution are 0.6462 and 0.7111. Hence, for the sum of 10 such observations, μ is 6.462 and σ is 2.248. Then

$$z = \frac{3.5 - 6.462}{2.248} = -1.32;$$

$$P'_{A,N} = P\left(\sum_{i=1}^{10} c_i \leq 3\right) \simeq .093,$$

while the exact multinomial probability is 0.089.

Consider, now, the state of the system when 10 consecutive lots are accepted under normal inspection, but the sum of the number of defectives in the 10 lots exceeds r^* in Table VIII. In the calculations of the next section, we will require two types of probabilities for these lots. First, the probability distribution for the total number of defectives in the 10 lots will be

$$(8) \qquad g_{10}(s') = g_{10}(s \mid s > r^*) = g_{10}(s) \Big/ \sum_{r^*+1}^{10c} g_{10}(s),\ s' = r^* + 1, \ldots, 10c.$$

Secondly, the probability distribution for the number of defectives in a random lot chosen from the 10 will depend on the numerical value of s':

$$(9) \qquad f(c'_i) = f(c_i \mid s') = f(c_i) \cdot g_9(s' - c_i) \Big/ \sum_{c_i=0}^{c} f(c_i) \cdot g_9(s' - c_i),\ c'_i = 0, 1, \ldots, c.$$

Continuing the previous numerical example, where $n = 50$, $c = 2$, $r^* = 3$, and the process quality is 1.5-percent defective, assume that 10 consecutive lots have been accepted, but that these lots contain a total of exactly 5 defectives. By using the reasoning which lead to Equations (6) and (9), we have

c_i	$f(c_i)$	$g_9(5 = c_i)$	Joint	$f(c'_i)$
0	0.4923	0.1697	0.0835	0.5743
1	0.3692	0.1403	0.0518	0.3562
2	0.1385	0.0729	0.0101	0.0695

3. MARKOV CHAIN SOLUTION OF THE INSPECTION SCHEME

The three inspection plans may be separately formulated as finite state, discrete parameter Markov chains [8]. Table 3 illustrates the state space and transition probability matrix for normal inspection. The entries, ρ_a, represent $P_{A,N}$, and

$$\rho_r = 1 - \rho_a.$$

This matrix may be partitioned as indicated into the canonical form

$$\left(\begin{array}{c|c} I & 0 \\ \hline R & Q \end{array}\right)$$

The experiencing of a lot rejection during normal inspection forces entry into one of the transition states provided for all permutations of one rejection in the most recent j lot inspections, $j=1, 2, 3, 4$. Four acceptances subsequent to a rejection cause transition back to the "4A" state. A second failure in five, or less, consecutive lot inspections forces absorption into tightened inspection.

Transition states, labeled "KA," represent K consecutive lot acceptances, $K=1, 2, \ldots 9$. From state 9A, we have indicated a probability, ρ_a^* of absorption into reduced inspection. This is the joint probability that a tenth lot is accepted and that the limit number of Table VIII is not exceeded:

$$\rho_a^* = P_{A,N} P'_{A,N}. \tag{10}$$

From state 9A, we enter either reduced inspection with probability ρ_a^*, state $AAAR$ with probability ρ_r, or state 10A with probability $1-\rho_a^*-\rho_r$. State 10A implies 10 consecutive lot acceptances with the total defectives in the 10 lots greater than the limit number of Table VIII.

State 10A is a transient, not an absorbing state, since it is possible with additional inspections to reject a lot (moving to state $AAAR$) or to reach the limit number, r^*, for the 10 most recent lots, moving to reduced inspection. These are the only possible exits from state 10A. The probability of emerging to reduced inspection has been designated in Table 3 as ρ_a^{**}, and to $AAAR$, $1-\rho_a^{**}$.

Calculation of Probabilities For State 10A

Entrance into state 10A for the first time implies that the total number of defectives in the latest 10 lots is $r^*+1, r^*+2, \ldots, 10c$. The probability distribution for these states is given by equation (8). Now, as a new lot is sampled, 1 of the initial 10 lots containing $0, 1, \ldots, c$ defectives is replaced with a lot containing $0, 1, \ldots, n$ defectives. If this new lot has more than c defectives, there is an exit from 10A to $AAAR$. This probability is simply ρ_r. Otherwise, the new lot is accepted, and we must examine the possible transitions from r^*+1, r^*+2, etc., to each other and to reduced inspection. A limited number of such transitions are possible, since the total number of defectives cannot change by more than $\pm c$ as a single lot is replaced.

The process of replacing the original 10 lots, 1 by 1, with new lots may be represented as a step-by-step alteration of the probabilities in a transition matrix whose columns and rows represent the number

of defectives in the "old" and "new" lots. We may illustrate this semi-Markov process with the numerical example presented in section 2 of this paper ($n=50$, $c=2$, $r^*=3$, process quality 1.5-percent defective). Assume that state $10A$ has just been entered for the first time. Then the possible states are 4, 5, . . ., 20 defectives in the 10 lots. Using exact multinomial probabilities, Equations (6), (7), and (8) yield, for those states with probability at least 0.001,

Number of defectives in 10 old lots										
4	5	6	7	8	9	10	11	12	13	14
0.117	0.166	0.191	0.182	0.145	0.098	0.056	0.028	0.011	0.004	0.001

Deletion of a lot randomly chosen from these 10 reduces the total number of defectives by 0, 1, or 2, and produces a new probability vector for states 2, 3, . . ., 18. To calculate these probabilities, we must examine each starting state, 4, 5, . . ., 20 individually, since $f(c_i')$, as shown in Equation (9), is a function of the exact number of defectives in the 10 lots. For example, using the numerical result calculated in section 2 for 5 defectives in 10 lots, we have

c_i'	$f(c_i')$
0	0.5743
1	0.3562
2	0.0695

These are precisely the conditional probabilities of moving from state 5 to 5, 4, and 3, respectively, after one lot has been deleted; when multiplied by 0.166, the prior probability for state 5, we have the contribution of prior state 5 to the probability vector for the number of defectives in 9 lots. The process must be repeated for each prior state.

The deleted lot will be replaced with a new unconditional lot with probability distribution

c	Poisson probability
0	0.4724
1	0.3543
2	0.1328
(Reject)	0.0405

Since the probability distributions for defectives in this new lot and in the nine old lots are independent, we may multiply them to form a transition matrix. A portion of the resulting matrix is shown below:

Defectives in new lot	Defectives in 9 old lots									
	2	3	4	5	6	7	8	9	10	11
0	0.003	0.018	0.067	0.090	0.096	0.083	0.058	0.032	0.015	0.005
1	0.002	0.016	0.050	0.067	0.072	0.062	0.043	0.024	0.011	0.004
2	0.001	0.006	0.019	0.025	0.027	0.023	0.016	0.009	0.004	0.001
(Reject)	(sum of this row = 0.0405)									

TABLE 3. *Normal inspection state space and transition probabilities*—Continued

States	Transition probabilities																					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Reduced 1	1																						
Tightened 2		1																					
10A 3	ρ_a^{**}															$1-\rho_a^{**}$							
9A 4	ρ_a^{*}		$1-\rho_a^{*}-\rho_r$													ρ_r							
8A 5				ρ_a												ρ_r							
7A 6					ρ_a											ρ_r							
6A 7						ρ_a										ρ_r							
5A 8							ρ_a									ρ_r							
4A 9								ρ_a								ρ_r							
3A 10									ρ_a							ρ_r							
2A 11										ρ_a									ρ_r				
1A 12											ρ_a										ρ_r		
RAAA 13		ρ_r							ρ_a														
ARAA 14		ρ_r											ρ_a										
AARA 15		ρ_r												ρ_a									
AAAR 16		ρ_r													ρ_a								
RAA 17		ρ_r											ρ_a										
ARA 18		ρ_r												ρ_a									
AAR 19		ρ_r													ρ_a								

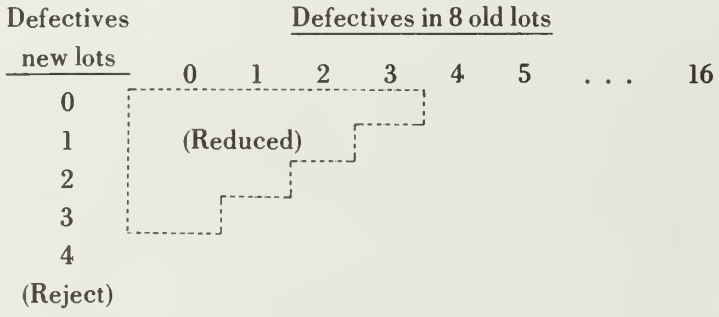
TABLE 3. *Normal inspection state space and transition probabilities* — Continued

RA 20		ρ_r															ρ_a					
AR 21		ρ_r																ρ_a				
R 22		ρ_r																		ρ_a		
Start 23											ρ_a										ρ_r	

The events enclosed by the broken line will lead to a transfer to reduced inspection, while the entire last row will lead to a transfer to normal inspection state *AAAR*.

We may now use the remainder of the transition matrix as a new prior state, delete a second old lot and add a second new lot. For elements in each column, recalculation of $f(c_i')$ is necessary using $g_8(s' - c_i)$, etc. The row probabilities are altered by applying unconditional Poisson probabilities for a single lot. For example, the prior state (1, 4) can transfer to (1, 4), (1, 3), (1, 2), (2, 4), (2, 3), (2, 2), (3, 4), (3, 3) (3, 2), the six states representing 0, 1, or 2 deletions, and 0, 1, or 2 additions.

The next matrix will have the appearance:



In general, the dimensionality of this matrix will begin at 1 by $10c + 1$, change to $c + 1$ by $9c + 1$, $2c + 1$ by $8c + 1$, etc., until the tenth deletion yields a $10c + 1$ by 1 array. After 10 deletions, this final column vector will represent the probability of states $r^* + 1$, $r^* + 2$, . . ., $10c$ for the beginning of a second cycle of deletions and additions. Calculations will be identical to those for the first cycle, except for the initial probability vector. Recycling may be continued until the initial probability vector for states $r^* + 1$, . . ., $10c$ contains negligible elements. At that point, the total probability of transfer to reduced inspection and to normal state *AAAR* will be available for insertion into the normal transition matrix (Table 3).

The expected number of lots processed per entry into state $10A$ may be readily computed from the probabilities of exiting to reduced inspection and normal state *AAAR* at each stage of deletion and addition. Thus, state $10A$ in Table 3 may be treated exactly the same as the other transient states in the normal transition matrix, except that, in any analysis of the expected number of lots processed under each transient state, we must multiply the number of entries into state $10A$ by the expected number of lots under "$10A$" inspection per entry.

Analysis of the Canonical Matrix

The expected transitions, $v_{i,j}$, for any starting state i and transient state j may be found by forming the fundamental matrix [8]

$$\mathbf{V} = \mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \ldots .$$

It may be shown that

$$\mathbf{V} = (\mathbf{I} - \mathbf{Q})^{-1}$$

and that the expected number of lots inspection during a single adoption of normal inspection is

$$E[m_n] = \sum_j v_{\text{start},j}.$$

When departure from normal inspection occurs, the probabilities, $b_{i,j}$, of absorption into state j, given starting state i, are found by

$$\mathbf{B} = \mathbf{V}\,\mathbf{R}.$$

For normal inspection, the probability of eventual absorption into reduced inspection is

$$a_{N,R} = b_{\text{start, reduced}} = v_{\text{start},9A}\,\rho_a^* + v_{\text{start},10A}\,\rho_a^{**},$$

and the probability of absorption into tightened inspection is

$$a_{N,T} = 1 - a_{N,R}.$$

TABLE 4. *Tightened inspection*

States		Transition probabilities					
		1	2	3	4	5	6
Normal	1	1					
4A	2	ρ_a					ρ_r
3A	3		ρ_a				ρ_r
2A	4			ρ_a			ρ_r
1A	5				ρ_a		ρ_r
Start	6					ρ_a	ρ_r

Table 4 shows the state space and transition probability matrix for tightened inspection. ρ_a represents $P_{A,T}$,

$$\rho_r = 1 - \rho_a.$$

States are provided for K consecutive lot acceptances, $K = 0, 1, \ldots, 5$, with $K = 5$ an absorption back to normal inspection, and $K = 0$ the starting state.

The expected number of lots inspected in an adoption of tightened inspection, $E[m_t]$, may be found for the maximum of 10 lots inspected on this plan by borrowing notation from the normal inspection model and determining

$$\mathbf{V} = \mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \ldots + \mathbf{Q}^9;$$

$$E[m_T] = \sum_j v_{\text{start},j}.$$

The probability of return to normal inspection is

$$a_{T,N} = b_{\text{start, normal}} = v_{\text{start, 4A}} \cdot P_{A,T}.$$

The probability of discontinuation of inspection is

$$1 - a_{T,N}.$$

TABLE 5. *Reduced inspection*

States		Transition probabilities 1	2	3
Normal/Accept	1	1		
Normal/Reject	2		1	
Start	3	ρ'_a	ρ_r	ρ_a

Reduced inspection transition probabilities and state space are shown in Table 5. ρ_a represents $P_{A,R}$, ρ'_a is $P'_{A,R}$, and

$$\rho_r = 1 - \rho_a - \rho'_a.$$

Using theory equivalent to that for normal inspection, the expected number of lots inspected for each adoption of reduced inspection may be simply determined as

$$E[m_R] = (1 - P_{A,R})^{-1}.$$

The probability of return to normal inspection with a rejected lot is

$$a_{R,N} = (1 - P_{A,R} - P'_{A,R})E[m_R],$$

$$a_{R,N} = (1 - P_{A,R} - P'_{A,R})/(1 - P_{A,R}).$$

For the duration of a MIL–STD–105D inspection scheme, the expected number of adoptions of normal, tightened, and reduced inspection may be obtained from the "meta" transition matrix shown in Table 6. Solution of this canonical matrix yields (assuming a start on normal inspection):

TABLE 6. *Meta-transition matrix*

States		1	2	3	4
Discontinue	1	1			
Tightened	2	$1 - a_{T,N}$		$a_{T,N}$	
Normal	3		$a_{N,T}$		$a_{N,R}$
Reduced	4			1	

Expected number of adoptions of	Equation
Reduced	$E[X_R] = a_{N,R}/(1 - a_{N,R} - a_{N,T}a_{T,N})$
Tightened	$E[X_T] = a_{N,T}/(1 - a_{N,R} - a_{N,T}a_{T,N})$
Normal	$E[X_N] = 1/(1 - a_{N,R} - a_{N,T}a_{T,N})$.

Then

$$E[l_R] = E[m_R]E[X_R],$$

$$E[l_T] = E[m_T]E[X_T],$$

$$E[l_n] = E[m_N]E[X_N].$$

4. SOME NUMERICAL RESULTS

Table 7 shows, for several *AQL* values and lot sizes, the results of submitting lots of consistent *AQL* quality to the MIL–STD–105D inspection scheme. We note from the Markov chain analysis that the probability of eventual discontinuation of sampling inspection is 1.0 if $p > 0.0$ since this is the final absorbing state in the Markov chains. The expected number of lots to absorption, the propor-

TABLE 7. *Some representative results from Markov solution of MIL–STD–105D inspection schemes when lots of* AQL *quality are submitted. Percentages shown are for the lots inspected up to the discontinuation point.*

AQL	Lot size	Expected number of lots before discontinuation	Expected percentage of lots on—			Expected percentage of lots rejected
			Normal	Tight	Red	
.25	90–1,200	107	74.2	25.8	0.0	21.5
	1,201–10,000	204	76.3	12.6	11.1	14.1
	10,001–35,000	763	63.5	3.2	33.3	4.9
1.00	16–150	104	73.4	26.6	0.0	22.0
	151–500	198	76.3	12.6	11.1	14.1
	501–1,200	671	65.2	3.6	31.2	5.3
	1,201–3,200	1,324	86.0	2.9	11.1	4.4
	3,201–10,000	5,465	85.1	0.6	14.3	1.7
2.50	9–50	107	74.2	25.8	0.0	21.5
	51–150	198	76.3	12.6	11.1	14.1
	151–280	666	65.2	3.6	31.2	5.3
	281–500	1,324	86.0	2.9	11.1	4.4
	501–1,200	5,465	85.1	0.6	14.3	1.7
	1,201–3,200	12,921	84.3	0.5	15.2	1.4
4.0	6–15	110	75.0	25.0	0.0	20.9
	16–50	179	76.1	13.9	10.0	15.2
	51–150	671	65.2	3.6	31.2	5.3
	151–280	1,125	82.1	3.2	14.7	4.7
	281–500	5,465	85.1	0.6	14.3	1.7
	501–1,200	8,623	86.6	0.6	12.8	1.7

tion of lots on normal, reduced, and tightened inspection, and the proportion of rejected lots has been indicated for each lot size. The results confirm the observation by Koyama [4] that switching rules in MIL–STD–105D are severe. Tightened inspection is frequent for small lot sizes, and the bulk of *AQL*-quality lots will undergo normal, rather than reduced, inspection.

A curious feature of MIL–STD–105D is that for some *AQL*'s the proportion on reduced inspection tends to be higher for small lots than for large lots. This is primarily a consequence of more stringent requirements to remain on reduced inspection. For example, Table IIC of MIL–STD–105D shows that, at $AQL = 1.0$, lots of size 1,200 require one or less defectives in 32 observations, while lots of size 3,200 require one or less defectives in 50 observations to remain on reduced inspection.

Table 8 shows the impact of incoming quality on the proportion of lots under normal, tightened, and reduced inspection for a particular (AQL, N) combination. The table reveals that more than one-half of the lots will be rejected, and sampling inspection will be discontinued in short order if p is twice the AQL. If p is half the AQL value, nearly all lots will be under reduced inspection, with virtually no rejections.

5. EXAMPLE OF COST CALCULATIONS

Consider the following problem. A manufacturer produces metal castings. The quality characteristic to be examined is the presence, or absence, of hot cracks to be fluoroscopically detected. Production is continuous and the manufacturer has some control, through negotiations with the purchaser, over lot size, which may vary from 50 to 550 castings.

The purchaser has specified MIL–STD–105D sampling with an Acceptable Quality Level of 4.0 percent defectives. Sampling inspection costs are estimated at \$1.50 per casting inspected. Rejected lots must be subjected to 100 percent screening, also at a cost of \$1.50 per casting.

The purchaser specifies that if sampling inspection is discontinued, the manufacturer must subject 10 subsequent lots to 100 percent screening in order to resume normal inspection again.

Suppose, for example, that the manufacturer produces material at the *AQL* level of quality (4.0 percent defectives), and submits lots of size 150. From MIL–STD–105D, the appropriate sampling plans are:

Type of sampling	n	c	r	r^*
Normal	20	2	—	4
Reduced	8	1	3	—
Tightened	20	1	—	—

TABLE 8. *Some results from Markov solution of MIL–STD–105D inspection with lot size at 150 and* AQL *at 4.0. Percentages shown are for the lots inspected up to the discontinuation point.*

Incoming quality, percent defective	Expected number of lots before discontinuation	Expected percentage of lots on—			Expected percentage of lots rejected
		Normal	Tight	Red	
2.0	664,666	12.8	0.0	87.2	0.2
3.0	8,800	36.8	0.5	62.7	1.1
4.0	671	65.2	3.6	31.2	5.3
5.0	154	78.0	11.6	10.4	15.3
6.0	69	74.4	23.0	2.6	28.8
7.0	45	65.0	34.5	0.5	41.7
8.0	35	55.5	44.4	0.1	52.2

The Markov chain analysis revealed that 671 lots are expected to be examined before discontinuation of sampling inspection, and that

431 lots are expected to be on normal inspection,
24 lots are expected to be on tightened inspection,
206 lots are expected to be on reduced inspection.

The expected number of rejected lots is 37. Hence, 47 lots out of 671 will be expected to be given 100 percent screening.

When these expected values are used, Equations (2), (3), and (4) yield:

$$E(f) = 0.107;$$

$$E(q) = 0.053;$$

$$E(c) = 1.50(0.107) + 1.50(0.053)$$

$$= \$0.24 \text{ per casting.}$$

Figure 1 shows the average inspection cost per unit, $E(C)$, as a function of lot size, N. The discontinuities in cost occur at the lot sizes where a new acceptance number and sample size take effect.

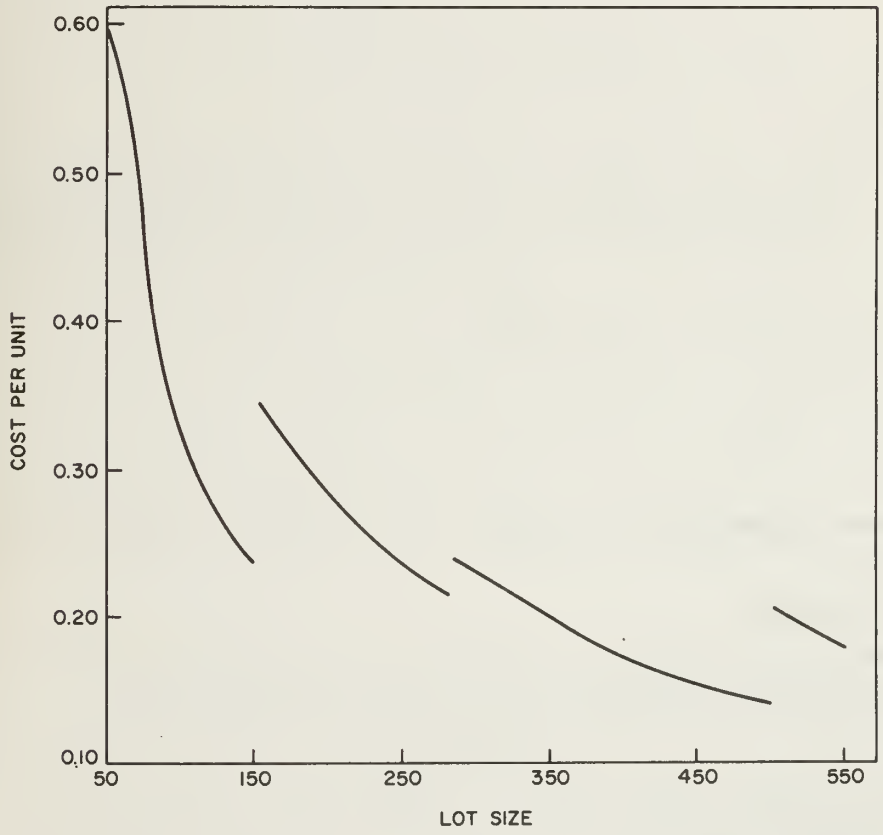


FIGURE 1. Cost per manufactured item as a function of lot size, $AQL = 4.0$, $p = AQL$.

These critical lot sizes are shown in Table 1 and are the same for normal, tightened, and reduced inspection in MIL–STD–105D.

Figure 1 shows that the set of interesting alternatives occurs in the vicinity of lot sizes equal to the upper boundaries shown in Table 1, namely 50, 90, 150, 280, 500. The manufacturer should produce in lots of 500, and the cost per casting will be \$0.14. Small lot sizes, say 60, could result in a cost per casting more than triple this figure.

Table 9 shows average cost per unit as a function of incoming quality, at a lot size of 150. Costs are extremely sensitive to p. Submission of 5 percent defective material, rather than 4 percent, to a 4.0 percent AQL plan causes a 67 percent increase in inspection costs.

TABLE 9. *Cost per Unit for Sampling Inspection with Lot Size 150* AQL = 4.0. *Calculations assume that rejected lots are screened 100 percent and that discontinuation of sampling requires screening of 10 lots.*

Incoming percent defective	Expected cost per unit
2.0	\$0.10
3.0	0.14
4.0	0.24
5.0	0.40
6.0	0.60
7.0	0.78
8.0	0.93

6. CONCLUSION

Analysis of these inspection schemes over realistic domains for AQL, p, and N reveals that the solution set will always occur in the vicinity of critical lot sizes as in the example. Thus, having estimated the costs of inspection for a product, a manufacturer need only perform this analysis for feasible lot sizes from the set 50, 90, 150, . . ., and in cases for which the maximum feasible lot size is not a critical lot size, this maximum should be examined as well.

The cost of inspection is very sensitive to lot quality, principally because to achieve and remain on reduced inspection very high quality lots must be submitted. Even when consistent AQL quality material is submitted to a MIL–STD–105D inspection scheme, a substantial number of lots may be subjected to tightened inspection when lot sizes are small. This implies that the switching rules in the document may be too severe, as suggested by other authors.

7. ACKNOWLEDGMENT

We are indebted to the referees and the editor for several valuable suggestions, which have improved the theoretical basis for the paper, and added clarity to the presentation.

REFERENCES

[1] Dodge, H. F. and H. G. Romig, *Sampling Inspection Tables* (John Wiley and Sons, New York, 2nd Edit. 1959).

[2] Grant, E., *Statistical Quality Control* (McGraw-Hill, Inc., New York, 3rd Edit. 1964).

[3] Hald, A., "The Compound Hypergeometric Distribution and a System of Single Sampling Inspection Plans Based on Prior Distributions and Costs," Technometrics, 2, 275–340 (Aug. 1960).

[4] Koyama, T., "Analysis on Dynamic Characteristics of Severity Control in MIL–STD–105D," International Conference on Quality Control, Tokyo, 1969.

[5] Mendizabal, P., "Ensayos Realizados Para Comprobar la Eficacia del Sistema MIL–STD–105D," Esayos e Investigacion, Vol. 1, No. 5, Bilbao, Spain (1967).

[6] *MIL–STD–105D 29 April, 1963 Military Standard—Sampling Procedures and Tables for Inspection* by *Attributes*. (United States Government Printing Office, Washington, D.C., 1963).

[7] Molina, E., *Poisson's Exponential Binomial Limit*, (Van Nostrand Company, Inc., New York, 1943).

[8] Parzen, E., *Stochastic Processes*, Holden-Day, San Francisco, 1962.

[9] *Tables of the Binomial Probability Distribution*, (United States Government Printing Office, Washington, D.C., 1952).

NEWS AND MEMORANDA

SUBSCRIPTION PRICE INCREASE

The United States Government Printing Office has raised the price of subscriptions to the Naval Research Logistics Quarterly to $10.00 per year in the United States and Canada, and $12.50 per year elsewhere. These prices reflect cost increases in the printing and mailing of all publications sold by the Government Printing Office.

Requests for new subscriptions should be sent to:

Superintendent of Documents
U.S. Government Printing Office
Washington, D.C. 20402

Checks should be made payable to *Superintendent of Documents.*

INDEX OF VOLUME 19

AGNEW, R. A., "Sequential Bid Selection by Stochastic Approximation," Vol. 19, No. 1, Mar. 1972, pp. 137–143.

BARON, D. P., "Quadratic Programming with Quadratic Constraints," Vol. 19, No. 2, Jun. 1972, pp. 253–260.

BELLMORE, M., J. C. Liebman and D. H. Marks, "An Extension of the (Szwarc) Truck Assignment Problem," Vol. 19, No. 1, Mar. 1972, pp. 91–99.

BENNINGTON, G., L. Gsellman and S. Lubore, "An Economic Model for Planning Strategic Mobility Posture," Vol. 19, No. 3, Sep. 1972, pp. 461–470.

BERGER, P. D., "Prediction with Zero-One Loss Structure," Vol. 19, No. 1, Mar. 1972, pp. 159–164.

BILES, W. E. and J. W. Schmidt, Jr., "A Note on a Paper by Houston and Huffman," Vol. 19, No. 3, Sep. 1972, pp. 561–567.

BRECHT, H. D. and A. C. Stedry, "Toward Optimal Bidding Strategies," Vol. 19, No. 3, Sep. 1972, pp. 423–434.

CABOT, A. V. and R. L. Francis, "Properties of a Multifacility Location Problem Involving Euclidian Distances," Vol. 19, No. 2, Jun. 1972, pp. 335–353.

CANDLER, W. and R. J. Townsley, "Quadratic as Parametric Linear Programming," Vol. 19, No. 1, Mar. 1972, pp. 183–189.

CHAN, L. K., S. W. H. Cheng and E. R. Mead, "An Optimum t-Test for the Scale Parameter of an Extreme-Value Distribution," Vol. 19, No. 4, Dec. 1972, pp. 715–723.

CHANG, R.C. and S. Ehrenfeld, "On a Sequential Test Procedure with Delayed Observations," Vol. 19, No. 4, Dec. 1972, pp. 651–661.

CHAUDHRY, M. L., "On the Discrete-Time Queue Length Distribution under Markov-Dependent Phases," Vol. 19, No. 2, Jun. 1972, pp. 369–378.

CHENG, S. W. H., L. K. Chan and E. R. Mead, "An Optimum T-Test for the Scale Parameter of an Extreme-Value Distribution," Vol. 19, No. 4, Dec. 1972, pp. 715–723.

CLARK, A. J., "An Informal Survey of Multi-Echelon Inventory Theory," Vol. 19, No. 4, Dec. 1972, pp. 621–650.

COHEN, E. A., JR. and K. D. Shere, "A Defense Allocation Problem with Development Costs," Vol. 19, No. 3, Sep. 1972, pp. 525–537.

COOK, F. X., A. S. Rhode and J. J. Gelke, "Impact of an All Volunteer Force upon the Navy in the 1972–1973 Timeframe," Vol. 19, No. 1, Mar. 1972, pp. 43–75.

CREMEANS, J. E. and H. S. Weigel, "The Multicommodity Network Flow Model Revised to Include Vehicle Per time Period and Node Constraints," Vol. 19, No. 1, Mar. 1972, pp. 77–89.

CUNNINGHAM, A. A. and A. L. Saipe, "Heuristic Solution to a Discrete Collection Model," Vol. 19, No. 2, Jun. 1972, pp. 379–388.

DAS, P., "Effect of Switch-Over Devices on Reliability of a Standby Complex System," Vol. 19, No. 3, Sep. 1972, pp. 517–523.

DEE, N. and J. C. Liebman, "Optimal Location of Public Facilities," Vol. 19, No. 4, Dec. 1972, pp. 753–759.

DELBROUCK, L. E. N., "The Law of Averages as a Computing Tool," Vol. 19, No. 1, Mar. 1972, pp. 149–158.

DERMAN, C., G. J. Lieberman and S. M. Ross, "On Optimal Assembly of Systems," Vol. 19, No. 4, Dec. 1972, pp. 569–574.

EHRENFELD, S. and R. C. Chang, "On a Sequential Test Procedure with Delayed Observations," Vol. 19, No. 4, Dec. 1972, pp. 651–661.

ERCAN, S. S., "Systems Approach to the Multistage Manufacturing Connected-Unit Situation," Vol. 19, No. 3, Sep. 1972, pp. 493–500.

FENNELL, J. and S. Zacks, "Bayes Adaptive Control of Two-Echelon Inventory Systems-I: Development for a Special Case of One-Station Lower Echelon and Monte Carlo Evaluation," Vol. 19, No. 1, Mar. 1972, pp. 15–28.

FINCHIM, J. A., JR., "Expectation of Contract Incentives," Vol. 19, No. 2, Jun. 1972, pp. 389–397.

FRANCIS, R. L. and A. V. Cabot, "Properties of a Multifacility Location Problem Involving Euclidian Distances," Vol. 19, No. 2, Jun. 1972, pp. 335–353.

GELKE, J. J., A. S. Rhode and F. X. Cook, "Impact of an All Volunteer Force upon the Navy in the 1972–1973 Timeframe," Vol. 19, No. 1, Mar. 1972, pp. 43–75.

GRINOLD, R. C., "The Payment Scheduling Problem," Vol. 19, No. 1, Mar. 1972, pp. 123–136.

GROSS, D. and A. Soriano, "On the Economic Application of Airlift to Product Distribution and Its Impact on Inventory Levels," Vol. 19, No. 3, Sep. 1972, pp. 501–507.

GSELLMAN, L., G. Bennington and S. Lubore, "An Economic Model for Planning Strategic Mobility Posture," Vol. 19, No. 3, Sep. 1972, pp. 461–470.

GUENTHER, W. C., "Tolerance Intervals for Univariate Distributions," Vol. 19, No. 2, Jun. 1972, pp. 309–333.

HABER, S. E. and R. Sitgreaves, "A Unified Model for Demand Prediction in the Context of Provisioning and Replenishment," Vol. 19, No. 1, Mar. 1972, pp. 29–42.

HABER, S. E., F. W. Segel and H. Solomon, "Statistical Auditing of Large-Scale Management Information Systems," Vol. 19, No. 3, Sep. 1972, pp. 449–459.

HANDLER, G., "Termination Policies for a Two-State Stochastic Process," Vol. 19, No. 2, Jun. 1972, pp. 281–292.

HECK, H. and S. Roberts, "A Note on the Extension of a Result on Scheduling with Secondary Criteria," Vol. 19, No. 2, Jun. 1972, pp. 403–405.
HENIN, C., "Computational Techniques for Optimizing Systems with Standby Redundancy," Vol. 19, No. 2, Jun. 1972, pp. 293–308.
HOLLOWAY, C. A., "A Mathematical Programming Approach to Identification and Optimization of a Class of Unknown Systems," Vol. 19, No. 4, Dec. 1972, pp. 663–679.
IJIRI, Y. and G. L. Thompson, "Mathematical Control Theory Solution of an Interactive Accounting Flows Model," Vol. 19, No. 3, Sep. 1972, pp. 411–422.
JARVIS, J. J. and J. B. Tindall, "Minimal Disconnecting Sets in Directed Multi-Commodity Networks," Vol. 19, No. 4, Dec. 1972, pp. 681–690.
KYDLAND, F., "Duality in Fractional Programming," Vol. 19, No. 4, Dec. 1972, pp. 691–697.
LEVY, E. and D. R. Limaye, "A Probabilistic Evaluation of Helicopter Lift Capability," Vol. 19, No. 4, Dec. 1972, pp. 761–775.
LIEBERMAN, G. J., C. Derman and S. M. Ross, "On Optimal Assembly of Systems," Vol. 19, No. 4, Dec. 1972, pp. 569–574.
LIEBMAN, J. C. and N. Dee, "Optimal Location of Public Facilities," Vol. 19, No. 4, Dec. 1972, pp. 753–759.
LIEBMAN, J. C., M. Bellmore and D. H. Marks, "An Extension of the (Szwarc) Truck Assignment Problem," Vol. 19, No. 1, Mar. 1972, pp. 91–99.
LIMAYE, D. R. and E. Levy, "A Probabilistic Evaluation of Helicopter Lift Capability," Vol. 19, No. 4, Dec. 1972, pp. 761–775.
LUBORE, S., G. Bennington and L. Gsellman, "An Economic Model for Planning Strategic Mobility Posture," Vol. 19, No. 3, Sep. 1972, pp. 461–470.
MARKS, D. H., M. Bellmore and J. C. Liebman, "An Extension of the (Szwarc) Truck Assignment Problem," Vol. 19, No. 1, Mar. 1972, pp. 91–99.
MASTROBERTI, R. and R. L. Nolan, "Productivity Estimates of the Strategic Airlift System by the Use of Simulation," Vol. 19, No. 4, Dec. 1972, pp. 737–752.
MEAD, E. R., L. K. Chan and S. W. H. Cheng, "An Optimum t-Test for the Scale Parameter of an Extreme-Value Distribution," Vol. 19, No. 4, Dec. 1972, pp. 715–723.
MILLHAM, C. B., "Constructing Bimatrix Games with Special Properties," Vol. 19, No. 4, Dec. 1972, pp. 709–714.
MORGENSTERN, O. and G. L. Thompson, "A Note on an Open Expanding Economy Model," Vol. 19, No. 3, Sep. 1972, pp. 557–559.
MORTON, T. E. and S. P. Sethi, "A Mixed Optimization Technique for the Generalized Machine Replacement Problem," Vol. 19, No. 3, Sep. 1972, pp. 471–481.
NOLAN, R. L. and R. Mastroberti, "Productivity Estimates of the Strategic Airlift System by the Use of Simulation," Vol. 19, No. 4, Dec. 1972, pp. 737–752.
PADBERG, M. W., "Equivalent Knapsack-Type Formulations of Bounded Integer Linear Programs: An Alternative Approach," Vol. 19, No. 4, Dec. 1972, pp. 699–708.
PARKER, J. B., "Bayesian Prior Distributions for Multi-Component Systems," Vol. 19, No. 3, Sep. 1972, pp. 509–515.
POSNER, M. J. M. and B. Yansouni, "A Class of Inventory Models with Customer Impatience," Vol. 19, No. 3, Sept. 1972, pp. 483–492.
RAVINDRAN, A., "Optimal Inventory Policies in Contagious Demand Models," Vol. 19, No. 1, Mar. 1972, pp. 191–203.
RHODE, A. S., J. J. Gelke and F. X. Cook, "Impact of an All Volunteer Force upon the Navy in the 1972–73 Timeframe," Vol. 19, No. 1, Mar. 1972, pp. 43–75.
ROBERTS, S. and H. Heck, "A Note on the Extension of a Result on Scheduling with Secondary Criteria," Vol. 19, No. 2, Jun. 1972, pp. 403–405.
ROODMAN, G. M., "Postoptimality Analysis in Zero-One Programming by Implicit Enumeration," Vol. 19, No. 3, Sep. 1972, pp. 435–447.
ROSS, S. M., C. Derman and G. J. Lieberman, "On Optimal Assembly of Systems," Vol. 19, No. 4, Dec. 1972, pp. 569–574.
SAATY, T. L., "Optimum Positions for m Airports," Vol. 19, No. 1, Mar. 1972, pp. 101–109.
SAHNEY, V. K., "Scheduling Data Transmission under an $\{S_1; O\}$ Policy," Vol. 19, No. 4, Dec. 1972, pp. 725–735.
SAIPE, A. L. and A. A. Cunningham, "Heuristic Solution to a Discrete Collection Model," Vol. 19, No. 2, Jun. 1972, pp. 379–388.
SALKIN, H. M., "A Note Comparing Glover's and Young's Simplified Primal Algorithms," Vol. 19, No. 2, Jun. 1972, pp. 399–402.
SCHMIDT, J. W., JR. and W. E. Biles, "A Note on a Paper by Houston and Huffman," Vol. 19, No. 3, Sep. 1972, pp. 561–567.
SEGEL, F. W., S. E. Haber and H. Solomon, "Statistical Auditing of Large-Scale Management Information Systems," Vol. 19, No. 3, Sep. 1972, pp. 449–459.
SETHI, S. P. and T. E. Morton, "A Mixed Optimization Technique for the Generalized Machine Replacement Problem," Vol. 19, No. 3, Sep. 1972, pp. 471–481.
SHERE, K. D. and E. A. Cohen, Jr., "A Defense Allocation Problem with Development Costs," Vol. 19, No. 3, Sep. 1972, pp. 525–537.
SILVER, E. A., "Inventory Allocation Among an Assembly and Its Repairable Subassemblies," Vol. 19, No. 2, Jun. 1972, pp. 261–280.
SIMON, R. M., "The Reliability of Multicomponent Systems Subject to Cannibalization," Vol. 19, No. 1, Mar. 1972, pp. 1–14.
SITGREAVES, R. and S. E. Haber, "A Unified Model for Demand Prediction in the Context of Provisioning and Replenishment," Vol. 19, No. 1, Mar. 1972, pp. 29–42.

SOLOMIN, H., S. E. Haber and F. W. Segal, "Statistical Auditing of Large-Scale Management Information Systems," Vol. 19, No. 3, Sep. 1972, pp. 449–459.
SORIANO, A. and D. Gross, "On the Economic Application of Airlift to Product Distribution and Its Impact on Inventory Levels," Vol. 19, No. 3, Sep. 1972, pp. 501–507.
SRINIVASAN, V. and G. L. Thompson, "An Operator Theory of Parametric Programming for the Transportation Problem—I," Vol. 19, No. 2, Jun. 1972, pp. 205–225.
SRINIVASAN, V. and G. L. Thompson, "An Operator Theory of Parametric Programming for the Transportation Problem—II," Vol. 19, No. 2, Jun. 1972, pp. 227–252.
SRINIVASAN, V. and G. L. Thompson, "Determining Optimal Growth Paths in Logistics Operations," Vol. 19, No. 4, Dec. 1972, pp. 575–599.
STEDRY, A. C. and H. D. Brecht, "Toward Optimal Bidding Strategies," Vol. 19, No. 3, Sep. 1972, pp. 423–434.
STONE, L. D., "Incremental Approximation of Optimal Allocations," Vol. 19, No. 1, Mar. 1972, pp. 111–122.
SWEAT, C. W., "A Duel Complicated by False Targets and Uncertainty as to Opponent Type," Vol. 19, No. 2, Jun. 1972, pp. 355–367.
TAYLOR, J. G., "On the Isbell and Marlow Fire Programming Problem," Vol. 19, No. 3, Sep. 1972, pp. 539–556.
THOMPSON, D. E., "Stochastic Duels Involving Reliability," Vol. 19, No. 1, Mar. 1972, pp. 145–148.
THOMPSON, G. L. and Y. Ijiri, "Mathematical Control Theory Solution of an Interactive Accounting Flows Model," Vol. 19, No. 3, Sep. 1972, pp. 411–422.
THOMPSON, G. L. and O. Morgenstern, "A Note on an Open Expanding Economy Model," Vol. 19, No. 3, Sep. 1972, pp. 557–559.
THOMPSON, G. L. and V. Srinivasan, "Determining Optimal Growth Paths in Logistics Operations," Vol. 19, No. 4, Dec. 1972, pp. 575–599.
THOMPSON, G. L. and V. Srinivasan, "An Operator Theory of Parametric Programming for the Transportation Problem—I," Vol. 19, No. 2, Jun. 1972, pp. 205–225.
THOMPSON, G. L. and V. Srinivasan, "An Operator Theory of Parametric Programming for the Transportation Problem—II," Vol. 19, No. 2, Jun. 1972, pp. 227–252.
TINDALL, J. B. and J. J. Jarvis, "Minimal Disconnecting Sets in Directed Multi-Commodity Networks," Vol. 19, No. 4, Dec. 1972, pp. 681–690.
TOWNSLEY, R. J. and W. Candler, "Quadratic as Parametric Linear Programming," Vol. 19, No. 1, Mar. 1972, pp. 183–189.
WEIGEL, H. S. and J. E. Cremeans, "The Multicommodity Network Flow Model Revised to Include Vehicle Per Time Period and Node Constraints," Vol. 19, No. 1, Mar. 1972, pp. 77–89.
WHEELER, A. C., "Stationary (s, S) Policies for a Finite Horizon," Vol. 19, No. 4, Dec. 1972, pp. 601–619.
YANSOUNI, B. and M. J. M. Posner, "A Class of Inventory Modles with Customer Impatience," Vol. 19, No. 3, Sep. 1972, pp. 483–492.
ZACKS, S. and J. Fennell, "Bayes Adaptive Control of Two-Echelon Inventory Systems-I: Development for a Special Case of One-Station Lower Echelon and Monte Carlo Evaluation," Vol. 19, No. 1, Mar. 1972, pp. 15–28.
ZIONTS, S., "Generalized Implicit Enumeration Using Bounds on Variables for Solving Linear Programs with Zero-One Variables," Vol. 19, No. 1, Mar. 1972, pp. 165–181.

U.S. GOVERNMENT PRINTING OFFICE: 1973— 525-459:8

INFORMATION FOR CONTRIBUTORS

The NAVAL RESEARCH LOGISTICS QUARTERLY is devoted to the dissemination of scientific information in logistics and will publish research and expository papers, including those in certain areas of mathematics, statistics, and economics, relevant to the over-all effort to improve the efficiency and effectiveness of logistics operations.

Manuscripts and other items for publication should be sent to The Managing Editor, NAVAL RESEARCH LOGISTICS QUARTERLY, Office of Naval Research, Arlington, Va. 22217. Each manuscript which is considered to be suitable material for the QUARTERLY is sent to one or more referees.

Manuscripts submitted for publication should be typewritten, double-spaced, and the author should retain a copy. Refereeing may be expedited if an extra copy of the manuscript is submitted with the original.

A short abstract (not over 400 words) should accompany each manuscript. This will appear at the head of the published paper in the QUARTERLY.

There is no authorization for compensation to authors for papers which have been accepted for publication. Authors will receive 250 reprints of their published papers.

Readers are invited to submit to the Managing Editor items of general interest in the field of logistics, for possible publication in the NEWS AND MEMORANDA or NOTES sections of the QUARTERLY.

NAVAL RESEARCH
LOGISTICS
QUARTERLY

MARCH 1973
VOL. 20, NO. 1
NAVSO P-1278

CONTENTS

ARTICLES

		Page
Structured Markovian Decision Problems	B. A. KALYMON	1
Polaroids: A New Tool in Non-Convex and in Integer Programming	C. BURDET	13
A Squared-VariableTransformation Approach to Nonlinear Programming Optimality Conditions	J. G. TAYLOR	25
Partial Information in Two Person Games with Random Payoffs	R. G. CASSIDY, C. A. FIELD, M. J. L. KIRBY	41
On Sequencing with Earliest Starts and Due Dates with Application to Computing Bounds for the (n/m/G /F_{max}) Problem	P. BRATLEY, M. FLORIAN, P. ROBILLARD	57
Scheduling Jobs, with Exponentially Distributed Processing Times, on Two Machines of a Flow Shop	A. A. CUNNINGHAM, S. K. DUTTA	69
Parametric Analysis of Linear Programs with Upper Bounded Variables	S. S. PANWALKAR	83
A Comparison of the Primal-Simplex and Complementary Pivot Methods for Linear Programming	A. RAVINDRAN	95
Applications of Mills' Differential	O. MOESCHLIN, G. BOL	101
An Efficient Heuristic Procedure for the Uncapacitated Warehouse Location Problem	B. M. KHUMAWALA	109
The Initial Provisioning Decision for Insurance Type Items	R. W. BURTON, S. C. JAQUETTE	123
Determination of the Optimal Investment in End Products and Repair Resources	M. ROSE	147
Requirements of an "Optimizer" for Computer Simulations	D. E. SMITH	161
A Cost Analysis of Sampling Inspection under Military Standard 105D	G. G. BROWN, H. C. RUTEMILLER	181
News and Memoranda		201
Index, Volume 19		203

OFFICE OF NAVAL RESEARCH
Arlington, Va. 22217